# UNRELATED PARALLEL MACHINES SCHEDULING UNDER MACHINE AVAILABILITY AND ELIGIBILITY CONSTRAINTS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES OF
ÇANKAYA UNIVERSITY

BY

ATIL KURT

IN PARTIAL FULFILMENT OF THE REQUREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

FEBRUARY 2012

Title of Thesis : **Unrelated Parallel Machines Scheduling Under Machine Availability and Eligibility Constraints**

Submitted by **Atıl KURT**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University

Prof. Dr. Taner ALTUNOK
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Nureddin KIRKAVAK
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science

Asst. Prof. Dr. Ferda Can ÇETİNKAYA
Director

**Examination Date** : 07.02.2012

**Examining Committee Members**

Prof. Dr. Meral AZİZOĞLU (METU)

Asst. Prof. Dr. Ferda Can ÇETİNKAYA (Çankaya Univ.)

Asst. Prof. Dr. Hakan GÜLTEKİN (TOBB ETU)

## STATEMENT OF NON-PLAGIARISM

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Atıl KURT

Signature : 

Date : 07.02.2012

# ABSTRACT

## UNRELATED PARALLEL MACHINES SCHEDULING UNDER MACHINE AVAILABILITY AND ELIGIBILITY CONSTRAINTS

KURT, Atıl

M.Sc., Department of Industrial Engineering

Supervisor: Asst. Prof. Dr. Ferda Can ÇETİNKAYA

February 2012, 73 Pages

In the literature of the parallel machines scheduling, it is generally assumed that all machines are continuously available for processing jobs and each job can be processed by any machine. However, these assumptions become unrealistic in some industrial environments. In this study, we consider the problem of scheduling *n* independent jobs on *m* unrelated parallel machines subject to machine availability and eligibility constraints, given the maximum continuous working time before the maintenance of each machine and the maintenance time. Our objective is to minimize the makespan, which is the time to complete the processing of all jobs. We consider both resumable and non-resumable jobs, and develop mathematical models and heuristic algorithm that obtain exact and near-optimal solutions, respectively, for both cases with multiple machine unavailability periods. Computational experiments are done to evaluate the performance of our solution methods in terms of both quality and time. The results show that the proposed heuristic algorithm finds near-optimal solutions in very short time.



***Keywords***: Parallel Machine Scheduling, Machine Availability, Machine Eligibility, Resumable and Non-Resumable Jobs, Makespan

# ÖZ

## MAKİNE KULLANILIRLIĞI VE ELVERİŞLİLİĞİ KISITLARI ALTINDA ÖZDEŞ OLMAYAN PARALEL MAKİNELERDE İŞ ÇİZELGELEMESİ

KURT, Atıl

Yüksek Lisans, Endüstri Mühendisliği Anabilim Dalı

Tez Yöneticisi: Yrd. Doç. Dr. Ferda Can ÇETİNKAYA

Şubat 2012, 73 Sayfa

Paralel makinelerin çizelgelenmesi literatüründe genellikle makinelerin her zaman işlem görmeye hazır olduğu ve her işin herhangi bir makinede işlem görebileceği varsayılır. Oysa ki, bazı imalat ortamlarında bu varsayımlar gerçek dışı kalmaktadır. Bu çalışmada, makine kullanılırlık ve elverişlilik kısıtları altında *n* tane işin *m* tane özdeş olmayan paralel makinelerde iş çizelgelemesi problemi, her bir makinenin bakım işleri öncesinde kesintisiz bir şekilde en çok çalışabileceği süre ve bakım işleri süresi bilindiği durum için ele alınmıştır. Amacımız, maksimum tamamlanma zamanını (tüm işlerin bitirilme süresini) enazlamaktır. İşlerin devam ettirilebilir ve ettirilemez olabildiği durumların her ikisi de ele alınmış ve her iki problemin optimum çözümü için karışık tamsayılı bir doğrusal programlama modeli ile sezgisel çözüm algoritmaları geliştirilmiştir. Ayrıca, bu yöntemlerin çözüm üretme performansları hem çözüm kalitesi hem de zaman yönünden sınanmıştır. Sonuçlar, önerilen sezgisel yöntemlerin çok kısa sürede optimal çözüme yakın çözümler bulduğunu göstermiştir.

***Anahtar Kelimeler***: Paralel Makine Çizelgelemesi, Makine Kullanilirliği, Makine Elverişliliği, Devam Ettirilebilir ve Ettirilemez İşler, Tüm İşlerin Bitirme Süresi

# ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my supervisor Asst. Prof. Dr. Ferda Can ÇETİNKAYA for his guidance leadership, interest, patience and encouragement during this thesis. It was a great chance and opportunity to work with him.

I am also grateful to Prof. Dr. Meral AZİZOĞLU, Asst. Prof. Dr. Hakan GÜLTEKİN and Asst. Prof. Dr. Nureddin KIRKAVAK for accepting to review this thesis and their valuable contributions.

I am always indebted to my family, I am especially grateful to my parents Behzat KURT and Sultan KURT for their endless and unconditional love. I also would like to thank my brothers Sadık KURT, Yusuf KURT and Volkan KURT, and my sister Binnaz KILIÇ for their support in my whole life. I am grateful to my cousin Kazım KURT for his supports during my educational life.

I would sincerely thank to all my friends who have contributed directly or indirectly to this thesis. I especially would like to extend my special thanks to Meltem BAYRAM for her friendship, helps and morale support throughout this study. I also thank to Mesut KOÇ for his help on my research.

It is great pleasure to be a member of IE assistants in Çankaya University, and I would like to thank all my friends at the department for their friendship. In my years of assistantship, they provide every technical and moral support to me.

Finally, I would like to thank TÜBİTAK (The Scientific and Technological Research Council of Turkey) for supporting this study through a graduate study scholarship.

# TABLE OF CONTENTS

# LIST OF TABLES

**TABLES**

# LIST OF FIGURES

**FIGURES**

# LIST OF ABBREVIATIONS

| | |
|---|---|
| **BFD** | Best Fit Decreasing |
| **BI** | Best integer |
| **CNC** | Computer Numerical Control |
| **CP** | Constraint Programming |
| **CPU** | Control Processing Unit |
| **FFD** | First Fit Decreasing |
| **GB** | Gigabyte |
| **GHz** | Gigahertz |
| **GoS** | Grade of Service |
| **IP** | Integer Programming |
| **LB** | Lower Bound |
| **LP** | Linear Programming |
| **LPR** | Linear Programming for Resumable |
| **LPT** | Longest Processing Time |
| **LS** | List Scheduling |
| **MBS** | Minimum-bin-slack |
| **MILP** | Mixed integer linear programming |
| **NP** | Non-deterministically Polynomial |
| **NR** | Non-resumable |
| **P** | Polynomial |
| **Pm** | Identical parallel machines environment |
| **Qm** | Uniform parallel machines environment |
| **R** | Resumable |
| **RAM** | Random Access Memory |
| **Rm** | Unrelated parallel machine environment |
| **SAW** | Sufficient average weight |
| **U** | Uniform distribution |
| **UB** | Upper bound |

# CHAPTER 1

# INTRODUCTION

Scheduling problem in the manufacturing environments is concerned with the optimal allocation or assignment of resources over time to a set of tasks. Resources are usually called *machines* and tasks are called *jobs*. Parallel machines scheduling problem is one of the well known scheduling problems where there is a set of *n* jobs to be processed exactly by one of the *m* machines functioning in parallel. Scheduling activity in this environment includes the assignment of *n* jobs to *m* parallel machines and sequencing of jobs on the assigned machines. Parallel machines scheduling environment is divided into three main categories. First category is *identical parallel machines* scheduling as each job has same processing time requirements on each machine. If the processing time of a job on a machine depends on the machine speed, then machine environment is defined as *uniform parallel machines*. General case of the parallel machine scheduling is *unrelated parallel machines* scheduling environment, in which processing time of jobs on each machine is different and independent.

Scheduling unrelated parallel machines is an important decision problem in the manufacturing and service industry. It aims to minimize production time and cost of manufacturing industry by effective use of staff and equipment. There are also examples in the service industry such as airlines and public transport. For the airlines example, the gates behave as unrelated parallel machines, and arriving and departing airplanes behave as jobs. Therefore, assigning hundreds of airplanes arriving and

departing daily to the gates will become an unrelated parallel machine scheduling problem.

Majority of the studies in the machine scheduling literature assume a stable machining environment in which all machines are continuously available for processing jobs throughout the scheduling period. However, this assumption may not be realistic in many industrial environments since machines may not be available during certain periods of time. *Machine unavailability* may occur due to machine breakdowns, preventive maintenances, tool changeovers, and material shortages.

There are many applications in the area about machine scheduling with unavailability constraints. For instance, suppose that an order has to be fixed in a time window. Therefore, this time window is an unavailability period for the scheduling of other jobs. Another example for the unavailability period is the scheduling CNC (computer numerical control) machines. CNC machines cannot be continuously available due to tool wear. So, it has to change its tool within a determined time. Machines are unavailable to process jobs during these tool change times.

Another example of machine unavailability is the case where the machines are not available due to preventive maintenance periods. Preventive maintenance activities (such as lubrication, cleaning, adjusting) reduce the machine breakdowns, and improve machine life. Therefore this leads to reduction of rework and scrap rate. Preventive maintenance is also an important issue for safety of workers and quality of products. Consequently, preventive maintenance is an effective activity to prevent the high cost of corrective maintenance which should be done after machine breakdowns.

Three types of job characteristics are discussed in the literature of scheduling problems with machine availability. These are resumable, non-resumable and semi-resumable jobs. In the *resumable jobs* case, if the processing of a job cannot be finished before an unavailability period of a machine, then its processing can be continued after the same machine becomes available again, i.e., the processing of a job is allowed to be interrupted and the interrupted job must continue to be processed on the machine it is interrupted on. In the *non-resumable jobs* case, job processing

must start and finish in the available period of the machine. In the *semi-resumable jobs* case, nonfinished job started before the down period can be continued with some extra setup time after the machine becomes available again.

Blazewicz et al. (2000) state that prescheduled urgent tasks may block the computer system. Therefore these prescheduled job times are unavailable for the processor. Computer systems continue processing the same task after unavailability periods, i.e., resumable.

A second well-known assumption used in the scheduling literature is that each job can be processed by any machine. But this assumption is not valid for some manufacturing environments like semiconductor manufacturing industry. Jobs may not be processed by any machine due to capabilities of machines. Sophistication and technological capabilities affect the processing of jobs so that jobs may have specific machine sets to be processed. This situation presents why the *machine eligibility* restriction is considered in this thesis.

An application of scheduling with eligibility restriction in a semiconductor manufacturing industry is introduced by Centeno and Armacost (1997). They state that complexity and miniaturization of integrated circuits needs more sophisticated equipment. Moreover, food processing plants can also be given as an example of eligibility restriction. In some service industries, customers are categorized and labeled with grade of service (GoS) levels. Therefore, the customers are served when the GoS level of the customer is no less than the GoS level of the server.

It is clear that the unrelated parallel machine scheduling problem with eligibility constraints is a special case of the classical unrelated parallel machine scheduling problem. Moreover, the identical and uniform parallel machines scheduling problems with eligibility constraint are also special cases of the unrelated parallel machines scheduling problem. Thus, any algorithm that solves the classical unrelated parallel machines problem also solves the problem of identical and uniform parallel machines scheduling problem with eligibility restriction.

The problem considered in this thesis is the scheduling of *n* independent jobs on *m* unrelated machines with machine availability and eligibility constraints to minimize the makespan, which is the time to complete the processing of all jobs. We assume that the unavailability of a machine is due to preventive maintenances or tool changes. We investigate the problem for both non-resumable and resumable jobs cases. To the best of our knowledge, there is no study that deals with scheduling problems with unrelated parallel machines under machine availability and eligibility constraints, and hence we intend to contribute to the scheduling literature in this direction.

The rest of this thesis is organized as follows. Chapter 2 briefly reviews the related work about the problem considered in this study. In Chapter 3 we define the problem, and develop mathematical models for both non-resumable and resumable jobs cases. We describe our proposed algorithm for solving the problems with non-resumable and resumable jobs and demonstrate the proposed algorithm by a numerical example in Chapter 4. The computational tests to evaluate the performance of the proposed heuristic algorithms and the mathematical models are presented in Chapter 5. Conclusions and several directions for future research are discussed in Chapter 6.

# CHAPTER 2

# LITERATURE REVIEW

In the literature of scheduling problems, it is commonly assumed that machines are continuously available and jobs are eligible to be processed by all machines. However, these assumptions are not always realistic in many manufacturing environments. Thus, this situation encourages the researchers to consider the machine availability and eligibility constraints. Although there are numerous studies considering machine availability and eligibility constraints independently, only a few studies consider these constraints simultaneously.

In this chapter, we briefly review the related work on the deterministic parallel machine scheduling with machine availability and eligibility constraints to minimize the makespan, which is the maximum completion time of the jobs. More details about the other scheduling criteria and machine environments for availability constraints, such as single machine, flow shops, and job shops can be seen in the literature review papers by Lee et al. (1997), Sanlaville and Schimdt (1998), and Ying et al. (2009). A review of the literature for the eligibility restriction is provided by Leung and Li (2008).

## 2.1 Parallel Machine Problems with Availability Constraints

Before examining the studies in the related literature, we want to introduce some unavailability period patterns which are classified in Figure 2.1.

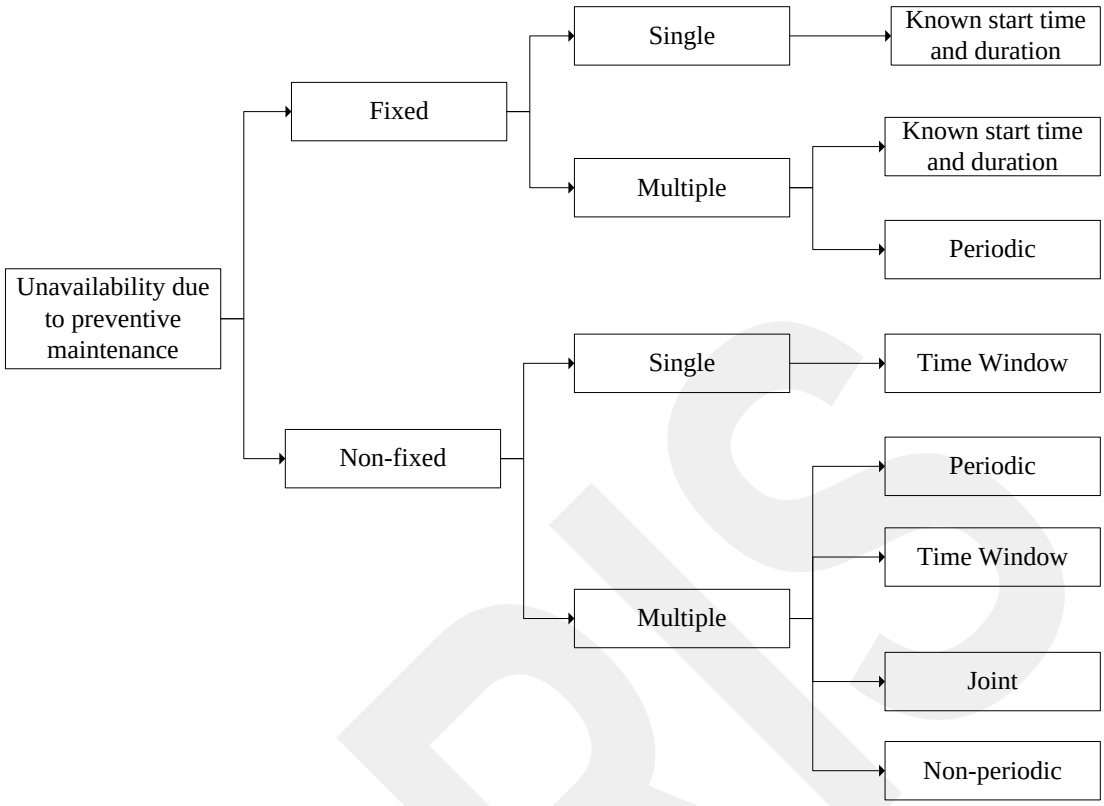


**Figure 2.1:** A Classification Scheme for Unavailability Due to Preventive Maintenance

In the literature, there are two categories based on the start time of the preventive maintenance activities: *fixed* and *non-fixed*. In the *fixed case*, the start time and duration of the unavailability period are known and constant. However, in the *non-fixed case*, job processing and maintenance activities are scheduled simultaneously so that the start time of the unavailability periods can change according to the schedule.

In the literature, some of the studies consider only one unavailability period on each machine (single), the other studies consider more than one unavailability periods (multiple). In the *single period* case, there is only one type of machine unavailability for both *fixed* and *non-fixed* cases. These are *known start time and duration*, and *time window* patterns, respectively. In the *known start time and duration* case, the start time of machine unavailability and its duration are fixed and known in advance. In the *time window* case, start time of a machine unavailability should be within a specified time window.

*Multiple fixed unavailability* case may have two different pattern types: *known start time and duration*, and *periodic*. In the first pattern type, the start time and duration of multiple unavailability periods are fixed and known in advance. In the second pattern type, the time between consecutive unavailability periods must be equal and constant, i.e., periodic.

*Multiple non-fixed unavailability* case may be *periodic*, *time window*, *joint*, and *non-periodic*. In the *periodic* case, the time between two consecutive unavailability periods should be less than or equal to a pre-determined constant time for each availability period. In the *time window* pattern, start time of each unavailability period should be within a specified time window. In the *joint* case, starting time and durations of unavailability periods depend on the machine's failure and repair rate. In the *non-periodic* pattern, the time between consecutive unavailability periods should be less than or equal to a pre-determined time which does not have to be same for each availability period.

Figure 2.2 represents the periodic patterns for the fixed and non-fixed unavailability cases.

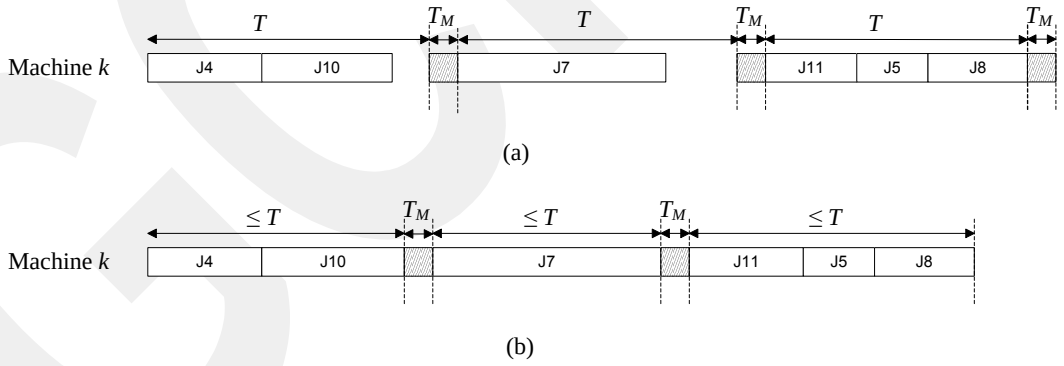


**Figure 2.2:** Periodic Patterns for the Fixed (a) and Non-Fixed (b) Unavailability Cases

In Figure 2.2, $T$ and $T_M$ represent the pre-specified availability and maintenance periods. Free space in Figure 2.2(a) represents the idle time on the machine. The

availability time between the two consecutive maintenances have to be equal to $T$ in the fixed case. However, it may be less than or equal to $T$ in the non-fixed case.

### 2.1.1 Non-resumable jobs case

In scheduling literature, problems with nonsimultaneous machine availability times are considered as a special case of the scheduling problem with machine availability constraints. In these problems, some machines may not be available at time zero although jobs are available. Lee (1991) studies the $m$ identical parallel machines scheduling problem with nonsimultaneous machine available times. He shows that the longest processing time (LPT) algorithm has a worst-case error bound less than $\left(3-\frac{1}{2m}\right) C_{\max}^{\text{opt}}$, where $C_{\max}^{\text{opt}}$ is the optimum makespan value. He also presents a modified LPT algorithm which has a tight error bound less than $\frac{4}{3} C_{\max}^{\text{opt}}$. Chang and Hwang (1999) study the same problem, and they apply the bin-packing algorithm MULTIFIT to solve the problem. They get better results rather than the Lee's modified LPT algorithm with a tight error of $\frac{2}{7}+2^{-k}$, where $k$ is the number of iterations in the MULTIFIT algorithm. Gharbi and Haouari (2005) study the same problem by considering job release dates and delivery times. They develop a branch-and-bound algorithm to find an optimal solution for the problem and report computational results with up to 700 jobs and 20 machines.

Lee (1996) considers the $m$ identical parallel machines problem by assuming that at least one machine is always available and other machines have one unavailable period. He analyzes two heuristic algorithms: List Scheduling (LS) and Longest Processing Time (LPT), and shows that the worst-case error bounds of LS and LPT are $m-1$ and $\frac{m+1}{2}$-1, respectively.

Hwang and Chang (1998) study the performance of LPT algorithm on $m$ identical parallel machines problem with the assumption that each machine has one unavailability period. They show that the worst-case error bound of the LPT

algorithm is 2 if more than half of the machines are available simultaneously and prove that this bound is tight. Hwang et al. (2005) generalize this bound by assuming arbitrary number of machines unavailable simultaneously. They prove that the worst case error bound of the LPT algorithm $1+\frac{1}{2}\left\lceil\frac{m}{m-\lambda}\right\rceil$ where $\lambda$ is the number of machines unavailable simultaneously.

Grigoriu and Friesen (2010) study the *m* identical parallel machines problem to minimize the makespan with the assumption that each machine has one unavailability period due to the shutdown of a machine. They propose an approximation algorithm called as LPTX, and show that worst-case error bound of this algorithm is $\frac{3}{2}$.

Liao et al. (2005) study the two identical machines problem with the assumption that only one machine has one unavailability period. They divide the problem into four sub-problems, and then each problem is solved optimally by an algorithm. The experimental results are also provided to evaluate the efficiency of their algorithm. Lin and Liao (2007) extend the two-machine problem to the case in which there is one unavailability period for both machines by assuming the same length of unavailability periods. The problem is solved optimally by an algorithm, and the computational experiments are reported for problems with up to 1000 jobs. It is observed that the algorithm finds optimum solution in very short CPU time and is quite efficient in solving large-scale problems.

Lee and Wu (2008) consider the *m* identical parallel machines with the deteriorating jobs for which job processing times are increasing functions of their starting times. They assume that each machine has one unavailability period. They propose a lower bound and a heuristic algorithm for the problem and discuss the performance of heuristic algorithm based on their computational experiments.

Xu et al. (2009) study the two identical parallel machines problem to minimize makespan with the assumption that only one machine is periodically unavailable. They show that LPT algorithm has a worst case error bound of $\frac{3}{2}$. Sun and Li (2010) extend this problem to the case where both machines have periodic unavailable times.

They propose the first fit decreasing (FFD) algorithm based on the one-dimensional bin-packing problem. Xu et al. (2008) extend the same problem to *m* identical parallel machines case in which each machine has periodic unavailability periods. They propose an approximation algorithm which is called as BFD-LPT. The BFD-LPT algorithm is based on the two popular algorithms. The first part of the name comes from Best Fit Decreasing (BFD) which is used for bin-packing problems. The second part of the name comes from the Longest Processing Time (LPT) which is used for the classical parallel machines problems. They also prove that problem is *NP*-hard, and unless $P = NP$, there is no polynomial time algorithm which has worst-case error bound of 2. Xu et al. (2010) modify the algorithm BFD-LPT to solve the same problem and analyze the performance of this modified algorithm.

Berichi et al. (2009) study the *m* identical parallel machines bi-objective problem to minimize both the makespan and the system unavailability time for the problem. Machines have multiple unavailability periods due to the joint production and maintenance activities. They develop five different heuristic algorithms to solve the problem. Their computational results show that their genetic algorithm developed based on the dominance properties gives efficient solutions.

All of the studies mentioned above consider the identical parallel machines. However, the other parallel machining environments (uniform and unrelated parallel machines) have not been investigated extensively. Yong (2000) proves that the LPT algorithm has a worst-case bound in the interval (1.52, 5/3) for the uniform parallel machines problem to minimize the makespan. Moreover, he develops an approximation algorithm, which has a worst case error bound of 6/5, for two machines problem. Suresh and Chaudhuri (1996) investigate the *m* unrelated parallel machine problem with multiple fixed unavailability periods. They develop a multi-pass heuristic algorithm with three different procedures named as lower bound, upper bound, and improve. In the improvement procedure, the solution obtained by the lower and upper bound procedures is improved by shifting and exchanging the jobs between two machines.

### 2.1.2 Resumable jobs case

In this section, we review the literature on parallel machines scheduling with unavailability constraints when the jobs are preemptive.

Lee (1996) studies the makespan minimization problem for $m$ identical parallel machines by assuming $m-1$ machines have one unavailability period and one machine is always available. He analyzes the performance of the LPT algorithm and modified LPT algorithm which is called as LPT2 (assigning a job on the top of the list to a machine such that the finishing time of that job is minimized). He also shows that the traditional LPT algorithm has an arbitrary large error bound, and LPT2 algorithm has a tight worst-case error bound of $\frac{1}{2}-\frac{1}{2m}$.

Liu and Sanlaville (1995) study the parallel machine problem with preemptive jobs by allowing that the number of available machines can change in time, and show that the makespan minimization problem is solvable in polynomial time.

Blazewicz et al. (2000) study the $m$ parallel machine scheduling problem with preemptive jobs in which machines have multiple unavailability intervals. They study the staircase pattern problem. Firstly, they show that intree case of the staircase pattern with preemption is *NP*-hard. For chain case, the identical parallel machine problem can be solved by a low-order polynomial time algorithm. They also prove that the unrelated parallel machine problem is NP-Hard for chain constraint. They also develop a network flow algorithm for the uniform parallel machines problem, and a two-phase linear programming based algorithm for the unrelated parallel machines problem.

Liao et al. (2005) consider the two identical machines problem with the assumption that only one machine has one unavailability period. They divide the problem into four cases, and then each case is solved optimally by an algorithm.

Lee and Wu (2008) study the $m$ identical parallel machines scheduling problem in which job processing times are increasing functions of their starting times and there is

one unavailability period for each machine. They propose a lower bound and a heuristic algorithm for the problem. Their computational experiments reveal that the performance of their heuristic algorithm and lower bound are satisfactory.

We prepare Table 2.1 to summarize the existing studies in the literature of makespan minimization on parallel machines scheduling with availability constraint.

## 2.2 Parallel Machine Problems with Eligibility Constraints

Machine scheduling with eligibility constraint is studied extensively in the scheduling literature under different names. In this section, we review the makespan minimization problems for parallel machines with eligibility restriction. More details about the other performance measures can be seen in the literature review paper by Leung and Li (2008).

Lenstra et al (1990) study the unrelated parallel machines scheduling problem which is a general case of the parallel machines scheduling problem with eligibility constraints. They develop a polynomial-time algorithm with a worst-case bound of 2. The same problem is also studied by Shchepin and Vaknaia (2005). They propose an algorithm, which has not been proved to be strongly polynomial, with an improved worst-case bound of 2-1/$m$.

Vairaktarakis and Cai (2003) consider the identical parallel machines problem with eligibility. They proposed four different heuristic algorithms and a branch-and-bound algorithm which can optimally solve the problem instances up to 50 jobs.

**Table 2.1:** Summary of the Existing Studies in the Literature of Parallel Machines Scheduling with Availability Constraints

| Machine Environment | Number of Unavailable Machines | Jobs case | Start time of unavailability periods | Number of unavailability periods | Type of unavailability patterns | Other job and machine characteristics | References |
|---|---|---|---|---|---|---|---|
| *Pm* | <*m* | NR | Fixed | Single | Known | Machine Release time | Lee(1991), Chang and Hwang (1999) |
| *Pm* | *m* | NR | Fixed | Single | Known | Machine and Job Release time, delivery date | Gharbi and Haouari (2005) |
| *Pm* | <*m*-1 | NR | Fixed | Single | Known | | Lee (1996) |
| *Pm* | *m* | NR | Fixed | Single | Known | | Hwang and Chang (1998), Hwang et al. (2005), Grigoriu and Friesen (2010) |
| *P*2 | 1 | NR | Fixed | Single | Known | | Liao et al. (2005) |
| *P*2 | 2 | NR | Fixed | Single | Known | | Lin and Liao (2007) |
| *Pm* | *m* | NR | Fixed | Single | Known | Deteriorating Jobs | Lee and Wu (2008) |
| *P*2 | 1 | NR | Fixed | Multiple | Period | | Xu et al. (2009) |
| *P*2 | 2 | NR | Fixed | Multiple | Period | | Sun and Li (2010) |
| *Pm* | *m* | NR | Fixed | Multiple | Periodic | | Xu et al. (2008), Xu et al. (2010) |
| *Pm* | *m* | NR | Non-fixed | Multiple | Joint | | Berichi et al. (2009) |
| *Qm* | *m* | NR | Fixed | Single | Known | | Yong (2000) |
| *Rm* | *m* | NR | Fixed | Multiple | Known | | Suresh and Chaudhuri (1996) |
| *Pm* | *m*-1 | R | Fixed | Single | Known | | Lee (1996) |
| *Pm* | <*m* | R | Fixed | Single | Known | | Liu and Sanlaville (1995) |
| *Pm* | *m* | R | Fixed | Multiple | Known | Chain, intree | Blazewicz et al. (2000) |
| *P2* | 1 | R | Fixed | Single | Known | | Liao et al. (2005) |
| *Pm* | *m* | R | Fixed | Single | Known | Deteriorating Jobs | Lee and Wu (2008) |
| *Pm* | *m* | R | Fixed | Multiple | Known | Eligibility | Liao and Sheen (2008) |

NR: Non-resumable R: resumable

Glass and Kellerer (2007) study some special cases of the identical parallel machines scheduling problem with eligibility constraints. They present an approximation algorithm to the case in which processing times of jobs are restricted to two values, 1 and $\lambda$, differing by at most 2, and show that worst-case bound of the algorithm is $2-\frac{1}{1+\lambda}$.

Edis and Ozkarahan (2011) consider a resource-constrained, identical parallel machine problem with eligibility constraints. They propose an integer programming (IP) model, a constraint programming (CP) model, and an IP/CP model which is a combination of IP and CP models. By using the computational results, they conclude that the performance of the IP/CP model is better than the performance of other two models.

There are three important special cases of the scheduling problems with eligibility constraint: The nested processing set, the inclusive processing set, and the equal-processing-time jobs. Glass and Kellerer (2007) define the *nested processing sets* as the sets that do not partially overlap and *inclusive processing sets* as the sets that are not nested but also include one another. If the processing times of jobs are equal, then it said to be *equal-processing-time jobs.*

Summary of the existing studies for the machine scheduling with eligibility constraints is given in Table 2.2.

**Table 2.2:** Summary of the existing studies for the machine scheduling with eligibility constraints

| Machine Environment | Other Job and Machine Characteristics | References |
|---|---|---|
| *Rm* | | Lenstra et al. (1990),Shchepin and Vakhania (2005) |
| *Pm* | | Vairaktarakis and Cai (2003), Glass and Kellerer (2007) |
| *Pm* | Resource-constrained | Edis and Ozkarahan (2011) |

## 2.3 Machine Scheduling with Availability and Eligibility Constraints

For the last two decades, many researchers have studied the parallel machines scheduling problem with machine availability and eligibility constraints independently. However, the existence of machine availability and eligibility constraints simultaneously in the same environment has not been investigated extensively in the literature. To the best of our knowledge, there is only one reported research for the problem that considers both availability and eligibility constraints simultaneously to minimize the makespan. Liao and Sheen (2008) consider *m* identical parallel machines scheduling problem in which each machine has multiple known fixed unavailability periods and resumable jobs have release times. In their study, the scheduling problem under consideration is transformed into a series of maximum flow problems, and solved by a polynomial time binary search algorithm.

In this thesis, we consider the unrelated parallel machines scheduling problem with machine availability and eligibility constraints. The most closely related study to ours is the study by Suresh and Chaudhuri (1996). In their study, it is assumed that there are multiple unavailability periods, the duration of unavailability periods are fixed and the start times of the unavailability periods are known. Our study differs from their study in the followings:

- We assume that the start time of unavailability periods in our study is non-fixed.
- The time between two consecutive unavailability periods should be less than or equal to a pre-determined constant time, i.e., periodic availability exists.
- We consider the eligibility restriction.

There are many studies in the scheduling literature that deal with makespan minimization problem in parallel machine environment; however, scheduling in an unrelated parallel machines with non-fixed periodic unavailability and machine eligibility constraints is studied first in this work to the best of our knowledge.

# CHAPTER 3

# PROBLEM DEFINITION AND MATHEMATICAL MODELS

In this chapter, we define the problem under consideration, and develop mathematical models for both non-resumable and resumable jobs cases.

## 3.1 Problem Definition

We consider the scheduling of independent jobs on unrelated parallel machines under machine availability and eligibility constraints with the following assumptions:

1. There are *n* jobs ready for processing at time zero.
2. There are *m* unrelated parallel machines which are available at time zero and the processing time of each jobs on each machine is different and independent.
3. Each job has only one operation to be processed.
4. Each job can only be processed on a set of machines.
5. Machines are not continuously available, and unavailability periods of the machines may overlap.

6. Maximum continuous working time $T_W$ and unavailability (down) time $T_U$ of each machine are constant and known.

7. No machine may process more than one job at a time

8. The processing time of each job on each machine is deterministic and known in advance.

For non-resumable jobs case, it is obvious that the maximum continuous working time of a machine is greater than or equal to the processing time of every job. Otherwise, no feasible solution is achieved. For resumable jobs case, the maximum continuous working time does not have to be greater than the processing time of every job.

We first present the following theorem that gives the property of an optimal schedule.

**Theorem 3.1.** *For both non-resumable and resumable jobs cases, there exists an optimal schedule in which the machine that determines the optimal makespan does not have any idle time. That is, machines are either processing a job or unavailable due to the maintenance activity.*

**Proof.** If an idle time exists on the machine that determines the makespan of a schedule, then subsequent jobs or unavailability periods may be moved earlier without increasing the makespan of the current schedule. This completes the proof. □

If jobs processed between two consecutive unavailability periods on a machine are considered to be a batch, then a schedule can be viewed as a series of batches of jobs separated by down periods on the machines. Figure 3.1 illustrates a feasible schedule for non-resumable jobs case on two machines.

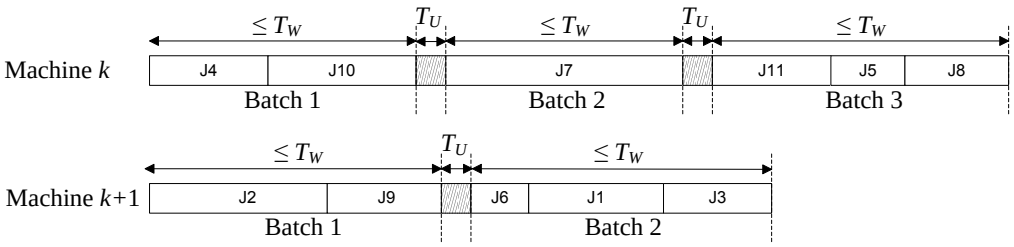


**Figure 3.1:** A Feasible Schedule for Non-Resumable Jobs Case as Batches of Jobs on Machine $k$ and $k$+1

As it is illustrated in Figure 3.1, the batch lengths may vary since the sum of the processing times of jobs in a batch may be different than the sum of the processing times of jobs in another batch may not be equal. Thus, the batch length shows the portion of the maximum continuous working time which has been used, and the number of batches gives the number of unavailability periods due to the maintenance activities.

**Theorem 3.2.** *For resumable jobs case, there exists an optimal schedule in which the length of each batch (except the last batch) of jobs on each machine is equal to the maximum working time* $T_W$.

**Proof.** If the length of a batch (except the last one) on a machine in a schedule is smaller than the maximum working time $T_W$, then some jobs or a portion of a job may be moved to the current batch so that the length of the current batch becomes equal to the maximum working time without increasing the makespan of the current schedule. As a result of consecutive movements on the same machine, there is a possibility of having a last batch with a batch length smaller than the maximum working time. This completes the proof. □

An example of a feasible schedule for resumable jobs case on two machines is shown in Figure 3.2.

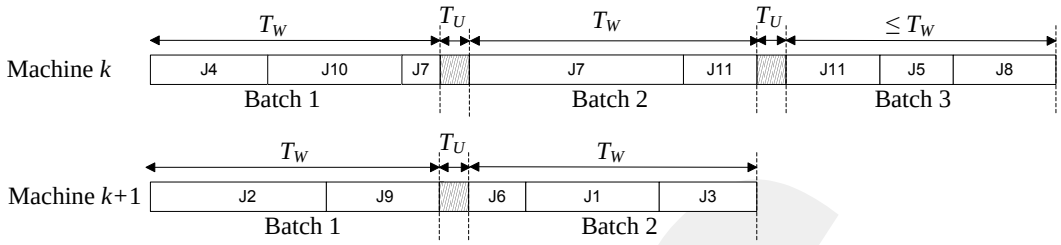


**Figure 3.2:** A Feasible Schedule for Resumable Jobs Case as Batches of Jobs on Machine $k$ and $k$+1

We observe that when machine availability and eligibility restrictions are omitted, the problem reduces to the scheduling of $n$ jobs on $m$ unrelated parallel machines to minimize the makespan, which is shown to be strongly *NP*-hard by Lenstra et al. (1990). Thus, our problem is also *NP*-hard in the strong sense.

## 3.2 Mathematical Models

Based on the assumptions and theorems given in the previous section, we now develop mixed integer linear programming (MILP) models of the problem for both non-resumable and resumable jobs cases.

### 3.2.1 Non-resumable jobs case

In this section, we propose a mixed integer linear programming (MILP) model to solve the problem optimally for non-resumable jobs case. Decisions considered for the non-resumable jobs case include: (i) how to allocate the jobs to the machines and (ii) how to allocate the jobs to the batches on the machines. We wish to find a feasible schedule that minimizes the maximum completion time (makespan) of all jobs. The following indices, sets, parameters and variables are used in this model.

***Indices and Sets*:**

$j$ Index for jobs ($j$=1,2,…, $n$)

$k$ Index for machines ($k$=1,2,…,$m$)

$M_j$ Set of eligible machines for job $j$

$J_k$ Set of jobs that can be processed by machine $k$

$t$ Index for batches ($t$=1,2,…,$\|J_k\|$, where $\|J_k\|$ is the number of jobs that can be processed by machine $k$)

***Parameters*:**

$T_W$ Maximum continuous working time for a machine

$T_U$ Unavailability (down) time

$P_{jk}$ Processing time of job $j$ on machine $k$ where $P_{jk}$ is set to a sufficiently large positive number if job $j$ is not eligible to be processed by machine $k$

***Decision Variables*:**

$X_{jkt} = \begin{cases} 1, & \text{if job } j \text{ is assigned to batch } t \text{ on machine } k \ (\forall j, k \in M_j, t = 1,2, \dots, \|J_k\|) \\ 0, & \text{otherwise} \end{cases}$

$Y_{kt} = \begin{cases} 1, & \text{if batch } t \text{ is used on machine } k \ (\forall k, \ t = 1,2, \dots, \|J_k\|) \\ 0, & \text{otherwise} \end{cases}$

$C_{max}$ = Makespan

***MILP Model*:**

The MILP model for the problem with non-resumable jobs (NRJ) can be formulated as follows:

(NR) Minimize $C_{max}$ (3.1)

Subject to

$$\sum_{k \in M_j} \sum_{t=1}^{\|J_k\|} X_{jkt} = 1 \qquad \text{for } j=1,2,\ldots,n \tag{3.2}$$

$$\sum_{j \in J_k} P_{jk} \times X_{jkt} \leq T_W \times Y_{kt} \qquad \text{for } k=1,2,\ldots,m;\ t=1,2,\ldots,\|J_k\| \tag{3.3}$$

$$\sum_{j \in J_k} \sum_{t=1}^{\|J_k\|} P_{jk} \times X_{jkt} + \left( \sum_{t=1}^{\|J_k\|} Y_{kt} - 1 \right) \times T_U \leq C_{max} \quad \text{for } k=1,2,\ldots,m \tag{3.4}$$

$$X_{jkt} \in \{0, 1\},\ Y_{kt} \in \{0, 1\} \text{ for } j=1,2,\ldots,n;\ k=1,2,\ldots,m;\ t=1,2,\ldots,\|J_k\| \tag{3.5}$$

In the above MILP model, objective (3.1) is to minimize the makespan. Constraint set (3.2) ensures that each job is assigned to one batch of a machine that is in the job's eligible set. Constraint set (3.3) guarantees that sum of the processing times of jobs in each batch of every machine cannot exceed the maximum continuous working time. The maximum completion time (makespan) of the jobs is given by the constraint set (3.4). Constraint sets (3.5) impose binary restrictions on the decision variables.

To improve efficiency of the mathematical model we introduce lower and upper bounds on the makespan value.

### 3.2.1.1 A lower bound on the makespan

A lower bound on the makespan is found by assuming that each job is assigned to its minimum processing time machine and the total work is equally allocated among these machines. Then we have

$$LB_1 = \max\left\{\frac{1}{m}\sum_{j=1}^{n}\min_{k\in M_j}\{P_{jk}\},\ \max_{j=1,\ldots,n}\left\{\min_{k\in M_j}\{P_{jk}\}\right\}\right\} + \left\lfloor\frac{\max\left\{\frac{1}{m}\sum_{j=1}^{n}\min_{k\in M_j}\{P_{jk}\},\ \max_{j=1,\ldots,n}\left\{\min_{k\in M_j}\{P_{jk}\}\right\}\right\}}{T_W}\right\rfloor \times T_U \quad (3.6)$$

where $\lceil x \rceil$ is the smallest integer greater than or equal to $x$, and $\lfloor y \rfloor$ is the largest integer smaller than or equal to $y$.

We add the following constraint to the mathematical model (NR).

$$C_{max} \geq LB_1 \quad (3.7)$$

### 3.2.1.2 An upper bound on the makespan

Instead of assuming an initial upper bound on the makespan as infinity, the makespan value obtained from the initial phase of the proposed heuristic algorithm in Section 5.1 can be used as an upper bound $UB_1$. i.e.,

$$UB_1 = \min\{C_{\max}^{S1},\ C_{\max}^{S2}\} \quad (3.8)$$

where $C_{\max}^{S1}$ and $C_{\max}^{S2}$ are the makespan of the initial schedules 1 and 2, respectively, obtained in the first phase of the proposed algorithm.

Thus, we add the following constraint to the mathematical model (NR).

$$C_{max} \leq UB_1 \quad (3.9)$$

In the modified MILP model, only one of the decision variables is a continuous variable, and other $\left(\sum \|J_k\|\right)(n+1)$ decision variables are binary. The model has $n + \sum \|J_k\| + m + 2$ constraints.

### 3.2.1.3 A special case with one eligible machine for each job

If we assume that only one machine is eligible for each job, then the problem reduces to the makespan minimization of *m* different single-machine scheduling problems with machine availability constraints only. In other words, we have *m* separate single-machine problems, and the machine having the longest completion time will determine the global makespan.

In placing the solution of this special case, one can solve *m* separate single-machine problems (equivalently, *m* separate *bin-packing problems* with a bin length of $T_W$) by the following mathematical model developed for any machine *k* and taking the completion time of the machine with the longest completion time as the global makespan. The single-machine problem for any machine *k* is stated below:

Minimize $\sum_{t=1}^{\|J_k\|} Y_{kt}$

Subject to

$$\sum_{t=1}^{\|J_k\|} X_{jt} = 1 \qquad \text{for } j \in J_k$$

$$\sum_{j \in J_k} P_{jk} \times X_{jt} \le T_W \times Y_{kt} \qquad \text{for } t = 1,2,\ldots,\|J_k\|$$

$$X_{jt} \in \{0, 1\},\ Y_{kt} \in \{0, 1\} \qquad \text{for } j \in J_k;\ t = 1,2,\ldots,\|J_k\|$$

In the above model, the objective is to minimize the total number of batches opened on machine $k$, which is equivalent to minimize the completion time on this machine. The first constraint set ensures that each job is assigned to one batch. The second constraint set guarantees that sum of the processing times of jobs in each batch cannot exceed the maximum continuous working time. The last constraint set impose binary restrictions on the decision variables.

The bin-packing problem is known to be *NP*-hard in the strong sense (Garey and Johnson 1972). Consequently, it is not expected to develop an exact solution method that can solve our problem in reasonable amount of time. To solve the bin-packing problem for assigning jobs to the batches of a machine, where each batch has a total processing capacity limited by the maximum continuous working time $T_W$, we use the heuristic algorithm SAWMBS proposed by Fleszar and Charalambous (2011) in our proposed algorithm since it is the most recent heuristic algorithm having the best performance among all available ones in the literature. Details of the SAWMBS algorithm are given in Appendix A.

### 3.2.2 Resumable jobs case

In the resumable jobs case, the processing of a job may continue after the same machine becomes available again if it cannot be finished before the unavailability period on this machine. That is, processing of the uninterrupted job will continue on the machine it is interrupted on. In this section, we propose the following mixed integer linear programming model to solve the problem with resumable jobs. In this formulation, the following decision variables are used:

$$X_{jk} = \begin{cases} 1, & \text{if job } j \text{ is assigned to machine } k \ (\forall j, k \in M_j) \\ 0, & \text{otherwise} \end{cases}$$

$B_k$ = Number of unavailability periods on machine $k$

$C_{max}$ = Makespan

The mixed integer linear programming model (R) for the resumable jobs case is described as follows:

(R) Minimize $C_{max}$ (3.10)

Subject to

$$\sum_{k \in M_j} X_{jk} = 1 \qquad \text{for } j=1,2,\ldots,n \qquad (3.11)$$

$$\frac{\sum_{j \in J_k} P_{jk} \times X_{jk}}{T_W} - 1 \leq B_k \qquad \text{for } k=1,2,\ldots,m \qquad (3.12)$$

$$\sum_{j \in J_k} P_{jk} \times X_{jk} + T_U \times B_k \leq C_{max} \qquad \text{for } k=1,2,\ldots,m \qquad (3.13)$$

$$X_{jk} \in \{0, 1\} \qquad \text{for } j=1,2,\ldots,n;\ k \in M_j \qquad (3.14)$$

$$B_k \geq 0 \text{ and integer} \qquad \text{for } k=1,2,\ldots,m \qquad (3.15)$$

In the above MILP model, objective (3.10) is to minimize the makespan. Constraint set (3.11) guarantees that each job is assigned to one machine which is in the job's eligibility set. Constraint set (3.12) determines the number of maintenance activities on each machine. The maximum completion time (makespan) of the jobs is given by the constraint set (3.13). Constraint sets (3.14) and (3.15) give binary and integrality restrictions on the decision variables, respectively.

In order to strengthen the mathematical model, we introduce the lower and upper bounds on the makespan, as it is presented in the following subsections.

### 3.2.2.1 A lower bound on the makespan

Consider the following linear programming (LP) model for the unrelated parallel machines problem with job preemption and machine eligibility without taking into

account the machine availability constraint. Note that in this problem all jobs are pre-emptable, i.e., the processing of a job is allowed to be interrupted and the interrupted job may continue to be processed on any machine (not necessarily on the machine it is interrupted on as in the resumable jobs case.).

Let $X_{jk}$ be portion of job $j$ assigned to machine $k$.

(U) Minimize $C_{max}$

Subject to

$$\sum_{k \in M_j} X_{jk} = 1 \qquad \text{for } j=1,2,\ldots,n \tag{3.16}$$

$$\sum_{j=1}^{n} P_{jk} \times X_{jk} \leq C_{max} \qquad \text{for } k=1,2,\ldots,m \tag{3.17}$$

$$\sum_{k \in M_j} P_{jk} \times X_{jk} \leq C_{max} \qquad \text{for } j=1,2,\ldots,n \tag{3.18}$$

$$X_{jk} \geq 0 \qquad \text{for } j \in J_k;\ k=1,2,\ldots,m \tag{3.19}$$

Let $C^u_{\max}$ be the optimal makespan of the above model. Then, a lower bound on the makespan for the original problem (R) given in (3.10) through (3.15) is

$$LB_2 = \left\lceil C^u_{max} + \left\lfloor \frac{C^u_{max}}{T_W} \right\rfloor \times T_U \right\rceil \tag{3.20}$$

**Theorem 3.3.** $LB_2 \geq LB_1$

**Proof.** From (3.17) and 3.18, the optimal makespan of the problem (U) is written as $C^u_{\max} = \max\left\{ \max_{k=1,\ldots,m}\left( \sum_{j=1}^{n} P_{jk} \times X_{jk} \right), \max_{j=1,\ldots,n}\left( \sum_{k \in M_j} P_{jk} \times X_{jk} \right) \right\}$. In the solution of the problem (U), if job $j$ is assigned to its minimum processing time machine without preemption, then we have

$$\sum_{k=1}^{m} P_{jk} \times X_{jk} = \min_{k \in M_j} \{P_{jk}\} \tag{3.21}$$

in (3.18).

On the other hand, if job *j* is not assigned to its minimum processing time machine or partially assigned among the machines, then we have

$$\sum_{k=1}^{m} P_{jk} \times X_{jk} > \min_{k \in M_j} \{P_{jk}\} \tag{3.22}$$

in (3.18) since the processing time of job *j* on the other machines is greater than the processing time on the machine having the minimum processing time for this job. Therefore, from (3.21) and (3.22), it is true that

$$\sum_{k=1}^{m} P_{jk} \times X_{jk} \geq \min_{k \in M_j} \{P_{jk}\} \tag{3.23}$$

for each job *j* in the solution of the problem (U).

Consider the following term in the lower bound $LB_1$ in (3.6):

$$\max\left\{\frac{1}{m}\sum_{j=1}^{n} \min_{k \in M_j} \{P_{jk}\}, \max_{j=1,\ldots,n}\left\{\min_{k \in M_j} \{P_{jk}\}\right\}\right\} \tag{3.24}$$

*Case* 1: $\frac{1}{m}\sum_{j=1}^{n} \min_{k \in M_j} \{P_{jk}\} \geq \max_{j=1,\ldots,n}\left\{\min_{k \in M_j} \{P_{jk}\}\right\}$

Summation of the terms in both sides of the inequality (3.23) for all jobs and then dividing both sides by $m$ yields

$$\frac{1}{m}\sum_{k=1}^{m}\sum_{j=1}^{n} P_{jk} \times X_{jk} \geq \frac{1}{m}\sum_{j=1}^{n} \min_{k \in M_j} \{P_{jk}\} \tag{3.25}$$

in the solution of the problem (U). Thus, $C_{\max}^{u} \geq \frac{1}{m}\sum_{j=1}^{n} \min_{k \in M_j} \{P_{jk}\}$.

*Case* 2: $\max_{j=1,\ldots,n}\left\{\min_{k\in M_j}\{P_{jk}\}\right\} > \frac{1}{m}\sum_{j=1}^{n}\min_{k\in M_j}\{P_{jk}\}$

From (3.23), we can write

$$\max_{j=1,\ldots,n}\left(\sum_{k=1}^{m} P_{jk}\times X_{jk}\right) \geq \max_{j=1,\ldots,n}\left(\min_{k\in M_j}\{P_{jk}\}\right) \tag{3.26}$$

in the solution of the problem (U). Thus, $C^{u}_{\max} \geq \max_{j=1,\ldots,n}\left(\min_{k\in M_j}\{P_{jk}\}\right)$.

From Cases 1 and 2, we conclude that $LB_2 \geq LB_1$ since $C^{u}_{max}$ is greater than or equal to the term $\max\left\{\frac{1}{m}\sum_{j=1}^{n}\min_{k\in M_j}\{P_{jk}\}, \max_{j=1,\ldots,n}\left\{\min_{k\in M_j}\{P_{jk}\}\right\}\right\}$ in the lower bound $LB_1$. □

### 3.2.2.2 An upper bound on the makespan

An upper bound on the makespan may also be calculated by assuming that each job is assigned to its minimum processing time machine among its eligible machines. Then we have

$$UB_2 = \max_{k=1,\ldots,m}\left\{\sum_{j=1}^{n}\min_{k\in M_j}\{P_{jk}\} + \left\lfloor\frac{\sum_{j=1}^{n}\min_{k\in M_j}\{P_{jk}\}}{T_W}\right\rfloor \times T_U, \max\left\{\min_{k\in M_j}\{P_{jk}\}\right\}\right\} \tag{3.27}$$

These lower and upper bounds are added to the mathematical model (R) with the following constraints:

$$C_{max} \geq LB_2 \tag{3.28}$$

$$C_{max} \leq UB_2 \tag{3.29}$$

### 3.2.2.3 A special case with one eligible machine for each job

When only one machine is eligible for each job, then the problem reduces to the makespan minimization of $m$ different single-machine scheduling problems with machine availability constraint only. Solution of this model is trivial since the jobs are resumable. Sequencing the jobs randomly on each machine and then making maintenance activities at every $T_W$ time units gives the minimum completion on each machine. That is, the optimal makespan is

$$C^*_{\max} = \max_{k=1,\ldots,m} \left\{ \sum_{j \in J_k} P_{jk} + \left\lfloor \frac{\sum_{j \in J_k} P_{jk}}{T_W} \right\rfloor \times T_U \right\} \tag{3.30}$$

Also, note that the optimal makespan of the general case is given by (3.30) if the assignments of the jobs are known in advance.

# CHAPTER 4

# PROPOSED ALGORITHMS

The size of the MILP models for non-resumable and resumable jobs cases discussed in the previous chapter increases drastically with the number of jobs. Therefore, the optimal solution to large-sized problems may not likely be obtained within reasonable computational times. Moreover, the existence of a polynomial-time algorithm to solve the problem optimally is unlikely since we have an *NP*-hard problem. This motivated us to develop a fast algorithm that provides near-optimal solutions within relatively very short times.

The composite algorithm proposed in this chapter is a multi-phase heuristic handling each job type cases (non-resumable and resumable) and involving four phases called *finding an initial schedule*, *improvement by shifting jobs to other machines, improvement by swapping (pairwise interchanging) jobs between machines,* and *building a new schedule by perturbing the current one.*

In the first phase of the algorithm, initial feasible schedules are generated. In the second phase, each of the initial feasible schedules is improved by shifting jobs from their current machines to each other machine. In the third phase, the jobs on different machines in the resulting schedule obtained by the second phase are pairwise interchanged in order to further improve the current solution. Finally, in the fourth phase, the schedule found in the third phase is improved by making a machine in the

set of eligible machines for a job ineligible to process this job. At the end, the algorithm gives improved schedules for various initial schedules. The best of the resulting schedules is selected. We call the algorithms as Heuristic-NR and Heuristic-R for the non-resumable and resumable cases, respectively.

## 4.1 Phase 1: Finding an Initial Schedule

### 4.1.1 Case 1: Non-resumable jobs

We generate two initial feasible schedules for non-resumable jobs case. If one of these schedules has a makespan value which is equal to the lower bound $LB_1$ in (3.6), then this initial schedule is optimal; otherwise, we continue with remaining phases described in Sections 4.2 through 4.4 for each initial schedule, and select the best of two schedules as the proposed schedule by the algorithm Heuristic-NR.

***Initial Schedule* 1:** The stepwise description of the procedure that determines the Initial Schedule 1 is as follows:

Step 1: (i) Repeat this step for each job *j* in set *J*. If job *j* has only one eligible machine, then assign job *j* to the eligible machine and calculate the workload of the eligible machine when job *j* is assigned to that machine; otherwise, assign job *j* to unassigned jobs set *U*.

(ii) Repeat this step for each job *j* in set *U*. Find the machine(s) having the minimum processing time for job *j*. If there is only one eligible machine having the minimum processing time for job *j*, then assign job *j* to this machine, and calculate the workload of the eligible machine when job *j* is assigned to that machine; otherwise (i.e., there are more than one eligible machine having the same minimum processing time for job *j*), select the machine having the minimum work load among the set of eligible machines having the same minimum processing time for job *j*,

assign job *j* to the selected machine, and calculate the workload of the machine for which job *j* is assigned.

Step 2: For each machine, apply the bin-packing heuristic algorithm SAWMBS to find an initial schedule of all jobs assigned, and calculate the associated completion time on this machine as

$$C_k = \sum_{j \in J_k} P_{jk} + (B_k - 1) \times T_U$$

where $B_k$ is the number of batches on machine $k$ determined by the bin-packing heuristic algorithm SAWMBS.

Step 3: Calculate the makespan of the initial schedule of all jobs by selecting the maximum completion time among all machines as

$$C_{\max} = \max_{k=1,\ldots,m} \{C_k\}$$

***Initial Schedule* 2:** We only present the first step of the procedure that determines the Initial Schedule 2 since the last two steps of the procedures that determine the Initial Schedules 1 and 2 are same.

Step 1: Repeat this step for each job *j* in set *J*. Find the machine(s) having the minimum processing time for job *j*. If there is only one eligible machine having the minimum processing time for job *j*, then assign job *j* to this machine; otherwise (i.e., there are more than one eligible machine having the same minimum processing time for job *j*), select the machine having the minimum index, assign job *j* to the selected machine.

### 4.1.2 Case 2: Resumable jobs

In this section, we propose four initial feasible schedules for solving the problem with resumable jobs. If one of these four initial schedules has a makespan which is

not equal to the lower bound $LB_2$ in (3.20), then we continue with remaining phases described in Sections 4.2 through 4.4 and select the best of four schedules as the proposed schedule by the algorithm Heuristic-R. Otherwise, the initial schedule developed in Phase 1 is the optimal one.

***Initial Schedule* 1:** This schedule is obtained by the first step of the procedure that determines the Initial Schedule 1 for non-resumable jobs case.

An LP relaxation of the MILP model RJ developed for the resumable jobs case discussed in Section 3.2.2 is obtained by removing the integrality restrictions in (3.14) and (3.15) and keeping the non-negativity restrictions in these constraints. The optimal solution of the relaxed model (LPR) may give two results: (1) all $X_{jk}$'s are binary, and (2) a subset of $X_{jk}$'s are binary. If the first case occurs, then the solution of the model (LPR) is optimal for the original problem. Otherwise, we generate the following next three initial feasible schedules.

Let

$$Z_{j,a}^{i} = \begin{cases} 1, \text{ if job } j \text{ is assigned to machine } a \text{ in the initial schedule } i \\ 0, \text{ otherwise} \end{cases}$$

***Initial Schedule* 2:** This schedule is obtained by assigning each job to the most efficient machine, which is eligible to process this job and has the minimum processing time for this job among all eligible machines. Below is the stepwise description of the procedure that determines the Initial Schedule 2:

Step 1: Repeat this step for all jobs.

(i) Assign job $j$ to machine $a$ where $P_{j,a} = \min_{k \in M_j}\{P_{jk}\}$.

(ii) Set $Z_{j,a}^{1}=1$.

Step 2: Calculate the makespan as

$$C^1_{\max} = \max_{k=1,\ldots,m} \left\{ \sum_{j \in J_k} P_{jk} Z^1_{j,k} + \left\lfloor \frac{\sum_{j \in J_k} P_{jk} Z^1_{j,k}}{T_W} \right\rfloor \times T_U \right\}$$

***Initial Schedule* 3:** This schedule is obtained by solving the model (LPR), and then

- assigning each job $j$ having only one assignment variable with a value of 1 (i.e., for job $j$, $X_{ja} = 1$ for $a \in M_j$ and $X_{jk} = 0$ for $k \in M_j$, $k \neq a$) to machine $a$, and,
- assigning each job $j$, having no assignment value of 1 (i.e., for job $j$, $0 \leq X_{jk} < 1$ for all $k \in M_j$) to its most efficient machine.

The steps of the procedure that determines the Initial Schedule 3 are described below:

Step 1: Repeat this step for all jobs.

(i) Assign job $j$ to machine $a$ where $a \in M_j$ if $X_{j,a}=1$.

(ii) Assign job $j$ to machine $a$ where $P_{j,a} = \min_{k \in M_j}\{P_{jk}\}$ if $0 \leq X_{jk} < 1$ for all $k \in M_j$.

Step 2: Calculate the makespan as

$$C^2_{\max} = \max_{k=1,\ldots,m} \left\{ \sum_{j \in J_k} P_{jk} Z^2_{j,k} + \left\lfloor \frac{\sum_{j \in J_k} P_{jk} Z^2_{j,k}}{T_W} \right\rfloor \times T_U \right\}$$

***Initial Schedule* 4:** The method of determining the Initial Schedule 4 is very similar to the one that determines the Initial Schedule 3. It is obtained by solving the model (LPR), and then

- assigning each job $j$ having only one assignment variable with a value of 1 to machine $a$, and
- assigning each job $j$ having no assignment value of 1 to the machine having the largest assignment value between zero and one.

Below is the stepwise description of the procedure that determines the Initial Schedule 4:

Step 1: Repeat this step for all jobs.

(i) Assign job $j$ to machine $a$ where $a \in M_j$ if $X_{j,a}=1$.

(ii) Assign job $j$ to machine $a$ where $X_{j,a}=\max_{k\in M_j}\{X_{jk}\}$ if $0 \le X_{jk}<1$ for all $k \in M_j$.

(iii)Set $Z^3_{j,a}=1$.

Step 2: Calculate the makespan as

$$C^3_{\max}=\max_{k=1,\ldots,m}\left\{\sum_{j\in J_k} P_{jk}\, Z^3_{j,k}+\left\lfloor\frac{\sum_{j\in J_k} P_{jk}\, Z^3_{j,k}}{T_W}\right\rfloor\times T_U\right\}$$

## 4.2 Phase 2: Improvement by Shifting Jobs to Other Machines

In this phase the initial feasible schedule generated in Phase 1 is improved by shifting jobs from their current machines to the machines in their eligible sets. Below is the algorithmic description of the proposed algorithm in Phase 2.

Step 1: Rank the machines in nonincreasing order of their completion times $C_k$.

Step 2: Select a machine $k$ according to this order.

(i) Arrange the jobs assigned to machine $k$ in nonincreasing order of their processing times $P_{jk}$, i.e., LPT list.

(ii) Select a job *j* from the LPT list. If there is only one machine in the set of eligible machines for job *j*, then consider a different job in the list; otherwise, consider each machine $k'$ ($k' \in M_j$ and $k' \neq k$) which is eligible to process job *j*. Temporarily assign job *j* to each machine $k'$ if the condition $C_{k'} + P_{jk'} < C_k$ is satisfied, and go to Step 2(iii). If the condition $C_{k'} + P_{jk'} < C_k$ is not satisfied for every machine $k'$, then go to Step 2(ii) to select a different job from the list.

(iii) If the jobs are non-resumable, then apply the bin-packing heuristic algorithm SAWMBS for every machine $k'$ on which job *j* is temporarily assigned; otherwise, go to Step (iv).

(iv) Calculate the associated temporary completion time on machine $k'$ as

$C_{k'} = \sum_{j \in J_{k'}} P_{jk'} + (B_k - 1) \times T_U$ for non-resumable jobs case

$C_{k'} = \sum_{j \in J_{k'}} P_{jk'} + \left\lfloor \frac{\sum_{j \in J_{k'}} P_{jk'}}{T_W} \right\rfloor \times T_U$ for resumable jobs case

(v) Among the machines on which job *j* is temporarily assigned, determine machine $k''$ having the smallest temporarily completion time.

(vi) If the temporary completion time of machine $k''$ is less than completion time of machine *k*, then keep the schedule in which job *j* is temporarily shifted to machine $k''$, and go to Step 1; otherwise, do not shift job *j* , and go to Step 2(ii) to select another job.

Step 3: If all jobs in the LPT list of machine *k* are considered, then go to Step 2 to select another machine. Repeat until all machines are considered.

## 4.3 Phase 3: Improvement by Swapping (Pairwise Interchanging) Jobs between Machines

This phase takes the schedule obtained in Phase 2 and aims to improve it by swapping jobs between machines. The steps of the proposed algorithm in Phase 3 are as follows:

Step 1: Rank the machines in nonincreasing order of their completion times $C_k$.

Step 2: Select two machines $k^1$ and $k^2$, where $k^1$ and $k^2$ are the first and the last machines of the list found in Step 1, respectively.

(i) Arrange the jobs assigned to machine $k^1$ in nonincreasing order of their processing times $P_{jk^1}$. Call this list as LPT(1).

(ii) Arrange the jobs assigned to machine $k^2$ in nonincreasing order of their processing times $P_{jk^2}$. Call this list as LPT(2).

Step 3: Starting from the beginning of the list LPT(1), select one job $j^1$. Similarly, starting from the beginning of the list LPT(2), select another job $j^2$.

(i) If both machines $k^1$ and $k^2$ are eligible to process jobs $j^1$ and $j^2$ and the condition

$$\max\left(C_{k^1}, C_{k^2}\right) > \max\left(C_{k^1}+P_{j^2k^1}-P_{j^1k^1}, C_{k^2}+P_{j^1k^2}-P_{j^2k^2}\right)$$

is satisfied, then temporarily exchange the jobs, and go to Step 3(ii); otherwise, go to Step 3 to select another pair of jobs.

(ii) If the jobs are non-resumable, then apply the bin-packing heuristic algorithm SAWMBS for both machines $k^1$ and $k^2$ and calculate the temporary completion times on machines $k^1$ and $k^2$; otherwise, calculate the temporary completion times on machines $k^1$ and $k^2$, using the formula in Step 2(iv) of Phase 2.

(iii) Determine the temporary makespan of the schedule of all jobs by calculating the maximum completion time among all machines as

$$C_{\max} = \max_{k=1,\ldots,m} \{C_k\}$$

(iv) If the temporary makespan is greater than the completion time of machine $k^1$, then do not exchange the jobs $j^1$ and $j^2$, and go to Step 3 until all job pairs are considered for swapping; otherwise, do not exchange the jobs, and go to Step 2 until all remaining machine pairs are considered.

## 4.4 Phase 4: Building a New Solution by Perturbing the Current Solution

The aim of the last phase of the heuristic algorithm is to improve schedule found in Phase 3 by making a machine in the set of eligible machines for a job ineligible to process this job. The steps of the algorithm for Phase 4 are as follows:

Step 1: If an improved solution is obtained, then consider this improved schedule and go to Step 2; otherwise, convert the set of eligible machines for the selected job determined in Step 2 to its original set, and repeat this step for all remaining jobs.

Step 2: From the most loaded machine, select a job with the longest processing time.

Step 3: Make the machine, on which the selected job is already assigned, ineligible to process the selected job (i.e., the machine is not in the set of eligible machines for the selected job), and go to Phase 1.

## 4.5 Numerical Example

In this section, a numerical example is discussed to demonstrate the proposed heuristic algorithm. Consider a three-machine problem in which the data for machine

eligibility sets and processing times for ten jobs are given in Table 4.1, where the empty entries indicate the non-eligibility of the machines. i.e., machine $k \notin M_j$ for job $j$. Suppose that the maximum continuous working time and the unavailability time are 18 and 2, respectively.

**Table 4.1:** Eligibility Sets and Processing Times

| Job | Machine 1 | Machine 2 | Machine 3 |
|---|---|---|---|
| 1 | - | 6 | 5 |
| 2 | - | - | 7 |
| 3 | 10 | 7 | 15 |
| 4 | - | 10 | 13 |
| 5 | 5 | - | - |
| 6 | 9 | 9 | 11 |
| 7 | 13 | - | 14 |
| 8 | 10 | 11 | 7 |
| 9 | 11 | 5 | 6 |
| 10 | 7 | 11 | 12 |

**Case 1: Non-resumable jobs.** The optimal schedule with a makespan of 28 time units is obtained by solving the mathematical model NR, and is illustrated by Figure 5.1.

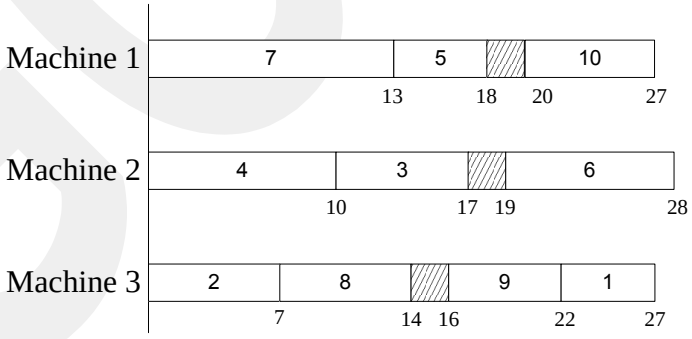


**Figure 4.1:** Optimal Schedule for the Non-Resumable Jobs Case

In this example, we observe that Initial Schedule 1 and 2 are same. The heuristic algorithms with Initial Schedule 1 or Initial Schedule 2 give the results below:

Phase 1: Step 1 finds the following feasible assignment of the jobs, which is given in Table 4.2.

**Table 4.2:** Job Assignments in Phase 1

| Machine | Jobs assigned | | | |
|---|---|---|---|---|
| 1 | 5 | 6 | 7 | 10 |
| 2 | 3 | 4 | 9 | |
| 3 | 2 | 1 | 8 | |

The application of the bin-packing heuristic for each machine yields the initial schedule 1 and 2 with a makespan of 36, as illustrated in Figure 4.2.

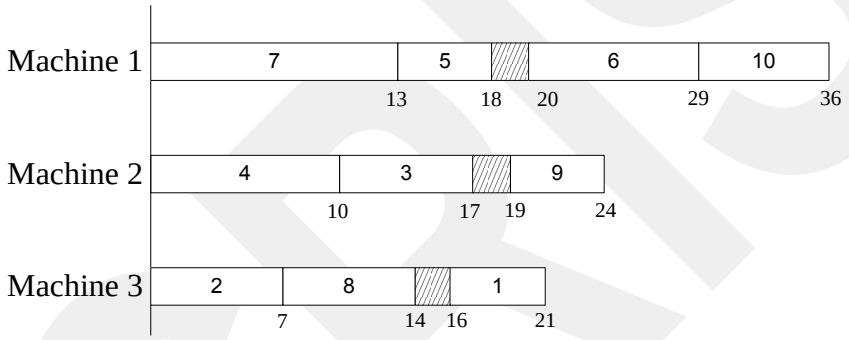


**Figure 4.2:** Schedule Obtained in Phase 1 (Non-Resumable Jobs Case)

Phase 2: The schedule obtained in Phase 1 is improved by shifting job 6 from machine 1 to machine 3, and shifting job 1 from machine 3 to machine 2. Phase 2 gives the schedule illustrated by Figure 4.3. The makespan of this schedule is 30, and the reduction in the makespan achieved by Phase 2 is 6 (=36−30).

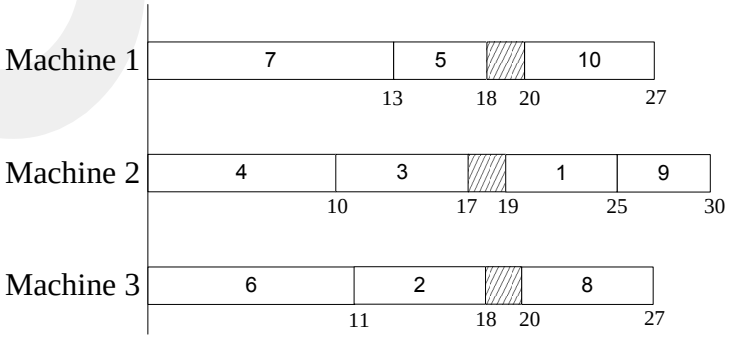


**Figure 4.3:** Schedule Obtained in Phase 2 (Non-Resumable Jobs Case)

Phase 3: The schedule obtained in Phase 2 is improved by swapping job 4 on machine 2 with job 6 on machine 3. Phase 3 gives the schedule illustrated by Figure 4.4, and the makespan of this schedule is 29.

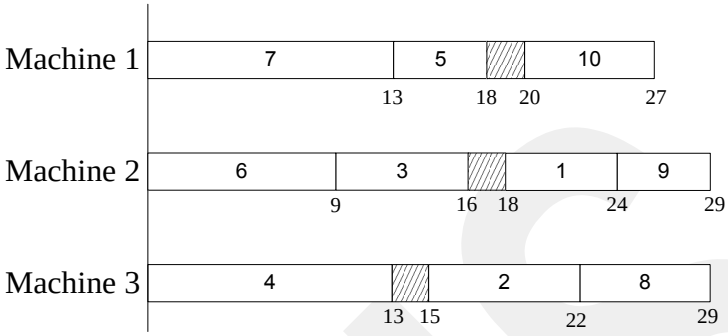


**Figure 4.4:** Schedule Obtained in Phase 3 (Non-Resumable Jobs Case)

Phase 4: The schedule obtained in Phase 3 is improved by making machine 2 ineligible to process job 1, which is already assigned to machine 2. Phase 4 gives the schedule shown in Figure 4.5, and the makespan of this schedule is 28.

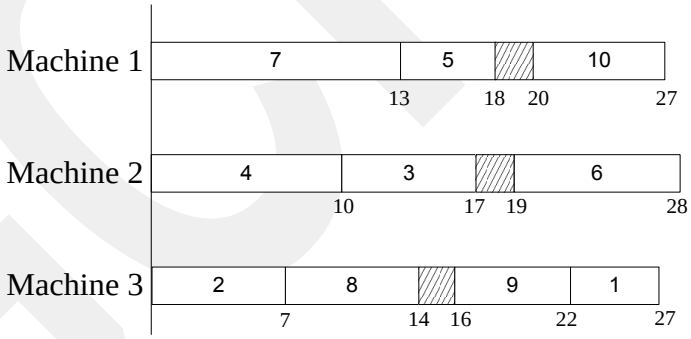


**Figure 4.5:** Schedule Obtained in Phase 4 (Non-Resumable Jobs Case)

Note that the makespan of the schedule obtained in Phase 4 by the proposed algorithm is equal to the makespan of the optimal schedule. Hence, by chance, our algorithm Heuristic-NR returns the optimal schedule given in Figure 4.1.

**Case 2: Resumable jobs.** A lower bound on the optimal makespan is obtained as

$$LB_2=28$$

The upper bound on the optimal makespan is 36 time units. The optimal schedule with a makespan of 28 time units is obtained by solving the mathematical model R with these lower and upper bounds, and is illustrated by Figure 4.6.

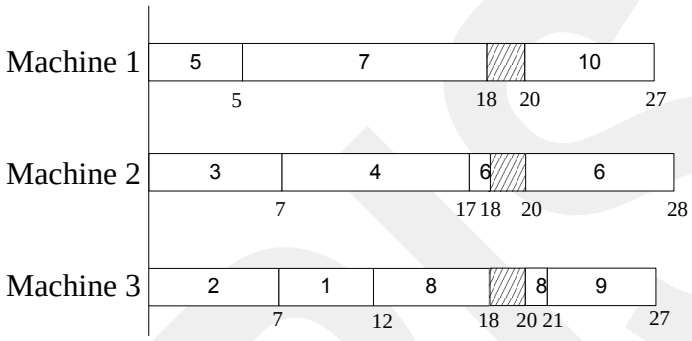


**Figure 4.6:** Optimal Schedule for the Resumable Jobs Case

The LPR model gives the following assignment in Table 4.3 for the problem.

**Table 4.3:** Assignments Obtained from the LPR Model

| Job | Machine 1 | Machine 2 | Machine 3 |
|---|---|---|---|
| 1 | - | - | 1 |
| 2 | - | - | 1 |
| 3 | - | 1 | - |
| 4 | - | 1 | - |
| 5 | 1 | - | - |
| 6 | 1 | - | - |
| 7 | - | - | 1 |
| 8 | 0.6 | 0.4 | - |
| 9 | - | 1 | - |
| 10 | 1 | - | - |

In this example, we observe that Initial Schedules 1, 2 and 3 are same. Therefore, the proposed algorithm Heuristic-R gives the following same results for these initial schedules.

Phase 1: The initial schedule with a makespan of 36 time units is obtained as shown in Figure 4.7.

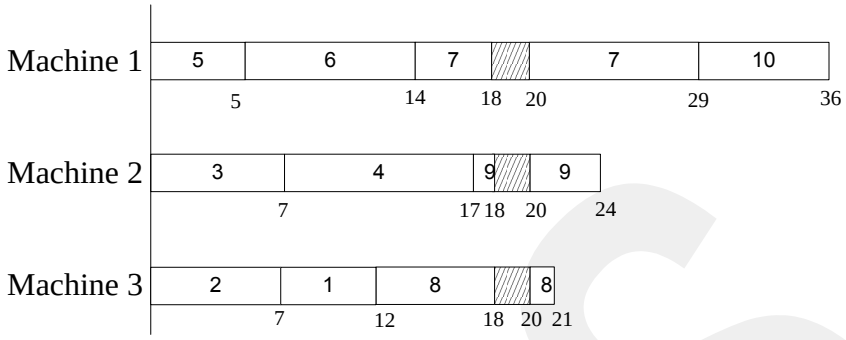


**Figure 4.7:** Schedule Obtained in Phase 1 (Resumable Jobs Case)

Phase 2: The schedule obtained in Phase 1 is improved by shifting job 7 from machine 1 to machine 3, and shifting job 8 from machine 3 to machine 1. Phase 2 gives the schedule illustrated by Figure 4.8. The makespan of this schedule is 33, and the reduction in the makespan achieved by Phase 2 is 3 (=36−33).

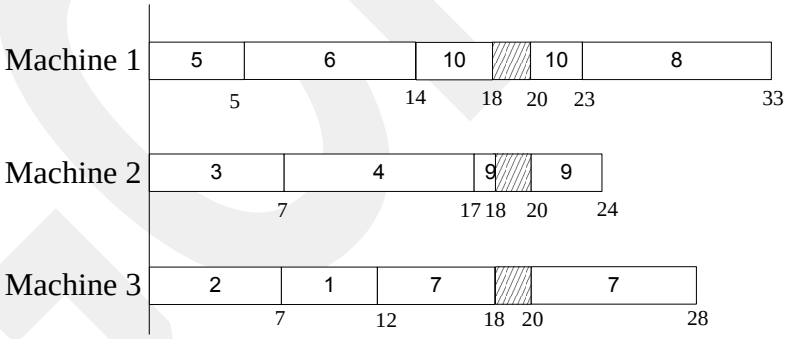


**Figure 4.8:** Schedule Obtained in Phase 2 (Resumable Jobs Case)

Phase 3: No improvement is achieved in the solution obtained in Phase 2.

Phase 4: The schedule obtained in Phase 3 is improved by making machine 3 ineligible to process job 7, which is already assigned to machine 3, and making

machine 2 ineligible to process job 1, which is already assigned to machine 2. Phase 4 gives the optimal schedule shown in Figure 4.6.

The algorithm Heuristic-R with Initial Schedule 4 gives the results below:

Phase 1: If we apply the Step 1 to the assignment given in Table 4.3 which is found by LPR model, we will find a feasible assignment of the jobs, which is given in Table 4.4

**Table 4.4:** Job Assignments in Phase 1 (with Initial Schedule 4 for Resumable Jobs Case)

| Machine | Job assigned | | | |
|---|---|---|---|---|
| 1 | 5 | 6 | 8 | 10 |
| 2 | 3 | 4 | 9 | |
| 3 | 1 | 2 | 7 | |

The Initial Schedule 4 with a makespan of 33 time units is obtained as shown in Figure 4.9.

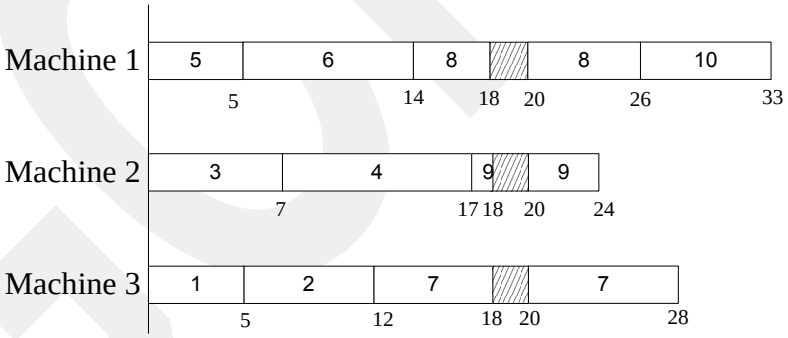


**Figure 4.9:** Schedule Obtained in Phase 1 (with Initial Schedule 4 for Resumable Jobs Case)

Phase 2: No improvement is achieved in the solution obtained in Phase 1.

Phase 3: No improvement is achieved in the solution obtained in Phase 2.

Phase 4: The schedule obtained in Phase 3 is improved by making machine 3 ineligible to process job 7, which is already assigned to machine 3, and making machine 2 ineligible to process job 1, which is already assigned to machine 2. Phase 4 gives the optimal schedule with a makespan of 28 as shown in Figure 4.6.

From the results obtained above, we may conclude that the algorithm Heuristic-R gives the optimal solution in this example.

# CHAPTER 5

# COMPUTATIONAL EXPERIMENTS

In this chapter, we describe our computational tests to evaluate the effectiveness and efficiency of the proposed heuristic algorithms in finding good quality schedules. For both non-resumable and resumable job cases, we compare the proposed algorithms with the mathematical models. The mathematical models are coded in GAMS 22.6 and solved by using CPLEX 11.0 under the time limit of 3 hours. The proposed heuristic algorithms are coded in Visual C++ 6.0. All computational experiments are conducted on a personal computer with Pentium Dual-Core IV 2.5 GHz CPU and 1 GB RAM under Windows XP operating system.

## 5.1 Computational Settings

The values of the parameters used in our experiments are generated as follows:

1. *Machine eligibility sets*: To set the machines eligible to process jobs, we adapt a similar method followed by Alagoz and Azizoglu (2003). For each machine and job combination, we generate a random number between 0 and 1. If this number is greater than 0.3, then the machine is assumed to be eligible to process the job and added to the eligibility set of this job. If no machine is eligible for this job, we generate another random number between 0 and 1 until an eligible machine is found.

2. *Processing times*: They are generated from discrete uniform distributions $DU(1, a)$, where $a$=10, 50. If a machine is not eligible for a job, then the processing time of this job is set to a sufficiently large positive number.
3. *Number of jobs and machines*: The numbers of machines are taken as 2, 3, 5, 10 and 20 whereas numbers of jobs are taken as 10, 20, 50, 100 and 200 by preserving the requirement that the number of jobs should be grater then the number of machines in a problem instance.
4. *Maximum continuous working time of the machines and unavailability times*: Maximum continuous working time $T_W$ is set to $a$, $2a$, and $3a$, whereas the unavailability time $T_U$ is set to $0.2T_W$, $0.5T_W$, and $1.0T_W$. We run our problem according to the combination of these maximum continuous working and unavailability times.

For each possible combination of the above parameters, 10 problem instances are generated. Hence, a total of 3,420 problems are tested for each of the non-resumable and resumable jobs cases.

## 5.2 Performance Measures

CPLEX gives two types of solutions for the MILP models. One of the solutions is the best integer solution which is the desired one; other solution is the best non-integer solution in which some of the variables are non-integer. If the best non-integer solution obtained is equal to the best integer solution, then we conclude that the optimal solution is achieved by the MILP model. Otherwise, the optimal solution is not found. For the problems with unknown optimal solutions, we compare the makespan obtained by our heuristic algorithm with the makespan of the non-integer solution, which is a lower bound on the optimal makespan value. In our experiments, we limit the runtime of the CPLEX for obtaining the optimal solution of each problem instance to 10800 seconds.

The following notation is used to measure the effectiveness of the solution approaches.

$C^B_{max}$= Makespan of the best integer solution obtained by the MILP model

$C^N_{max}$= Makespan of the non-integer solution obtained by the MILP model

$C^H_{max}$= Makespan of the solution obtained by a heuristic algorithm

$PE^M$= Percent error of the MILP solution

$$PE^M = \frac{C^B_{max} - C^N_{max}}{C^N_{max}} \times 100 \tag{5.3}$$

$PE^H$= Percent error of the solution obtained by a heuristic algorithm

$$PE^H = \frac{C^H_{max} - C^N_{max}}{C^N_{max}} \times 100 \tag{5.4}$$

Note that $C^N_{max} \leq C^B_{max}$, and $C^N_{max} = C^B_{max}$ at the optimal solution.

The efficiency measure of the proposed algorithms and the MILP models is the computational time required to solve the problem.

We also consider the number of optimum solutions obtained to evaluate the performance of solution approaches. The number of optimum solutions obtained is reported for problems where an optimal solution is obtained, i.e., $C^N_{max} = C^B_{max}$.

For some problem instances, an integer solution cannot be found by solving the MILP model. In order to find the number of unsolved problem instances by the MILP model, we use the number of non-optimum integer solutions obtained. The remaining problem instances show the number of unsolved problems.

To evaluate the effects of phases of heuristic algorithms, we consider the percent improvement on the makespan from one phase to another. Percent improvement obtained by a phase is calculated by the following formulas:

$$PI^{p_{i+1}} = \frac{C^{p_i}_{max} - C^{p_{i+1}}_{max}}{C^i_{max}} \tag{5.5}$$

where $C^{p_i}_{max}$ is the makespan value obtained at the end of Phase $i$, $i$ = 1, 2, 3, 4.

## 5.3 Discussion of the Results

In this section, the performance of solution approaches, and the effects of the phases of each heuristic algorithm developed for both non-resumable and resumable jobs cases are discussed.

### 5.3.1 Case 1: Non-resumable jobs

#### 5.3.1.1 Performance of the solution approaches

We now discuss the performance of the solution approaches which are given in Chapter 4. Firstly, we examine the performance of solution approaches with respect to the number of jobs $n$ and the number of machines $m$ when the maximum working time $T_W$ and unavailability time $T_U$ of the machines are fixed. Then, we will discuss the effects of $T_W$ and $T_U$ values to the performance of the solution approaches.

Computational results reveal that the change in the number of jobs and machines show nearly same characteristics on the performance of the solution approaches at each $T_W$ and $T_U$ combinations. We use Table 5.1 for problem set with $p_j \sim U[1, 10]$, to indicate the effects of a change in the number of jobs and machines when the $T_W$ is 10 and $T_U$ is 2. For the other $T_W$ and $T_U$ combinations of problem set with $p_j \sim U[1, 10]$, the performance of the solution approaches can be seen from Tables B-1 to B-8 given in the appendix to observe the effects of change in the number of job and machines. Moreover, we use Table 5.2 for problem set with $p_j \sim U[1, 50]$, to illustrate the effects of change in the number of job and machines when the $T_W$ is 50 and $T_U$ is 10. For the other $T_W$ and $T_U$ combinations of same sets, the performance of the solution approaches can be seen from Tables B-9 to B-16 to observe the effects of changes in the number of jobs and machines. From Tables 5.1, 5.2, B-1 to B-16, we observe that the heuristic algorithm Heuristic-NR with Initial Schedule 1 finds the best solution for 3214 (out of 3420) problem instances. However, the heuristic algorithm Heuristic-NR with Initial Schedule 2 finds the best solution for 2556 problem instances. This

implies that the performance of the Initial Schedule 1 is better than that of Initial Schedule 2.

The first set in Table 5.1 and 5.2 indicates the performance of the MILP, the next set illustrate the performance of Heuristic-NR. We discuss the performance of solution approaches by using both solution quality and CPU time of the algorithms. As can be seen from Table 5.1 and 5.2, the number of jobs has a significant effect on the maximum and average CPU times for the algorithms. The CPU time of the solution approaches increase rapidly as the number of jobs increase. Moreover, tables provide the CPU time of the best integer solution obtained by the MILP. From the table, we can conclude that the CPU time of the best integer solution is approximately equal to the average CPU time of small sized problems, and it is significantly smaller than the CPU of the large sized problems. In the MILP approach, some problems cannot be solved optimally in three hours. The CPU time of the MILP slightly decreases when the number of machines increases for the problems with $p_j \sim U[1, 10]$ and it does not show any systematic behavior when the number of machines increases for the problems with $p_j \sim U[1, 50]$. Furthermore, the number of machines has a little increasing effect on the CPU time of the heuristic algorithm.

In Tables 5.1 and 5.2, number of optimum solutions obtained by each solution approach is also reported. For the other problems, we do not have any information about the optimality. So we can conclude that these columns illustrate the minimum number of optimum solutions obtained by the solution approaches. Number of jobs has significant effect on the number of optimum solutions obtained. The number of optimum solutions decreases as the number of jobs increases. In the MILP, it is due to the fact that total number of constraints and variables increases as the number of jobs increases.

**Table 5.1:** Performance of the Solution Approaches for the Non-Resumable Jobs Case When $T_W$ is 10 and $T_U$ is 2

| | | MILP | | | | | | | Heuristic-NR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non-optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 1.63 | 2.06 | 1.63 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 10 | 0.00 | 0.00 |
| | 20 | 1.62 | 1.83 | 1.62 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 3242.29 | 10801.86 | 2.54 | 7 | 3 | 0.45 | 2.31 | 0.04 | 0.06 | 7 | 0.45 | 2.31 |
| | 100 | 4849.02 | 10801.89 | 11.91 | 6 | 4 | 0.26 | 0.76 | 0.39 | 0.60 | 5 | 0.33 | 1.04 |
| | 200 | 7751.32 | 10802.34 | 792.90 | 3 | 6 | 0.22 | 0.51 | 1.88 | 8.64 | 4 | 0.16 | 0.39 |
| 3 | 10 | 1.53 | 1.61 | 1.53 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.02 | 10 | 0.00 | 0.00 |
| | 20 | 1.61 | 1.70 | 1.61 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 9 | 0.59 | 5.88 |
| | 50 | 2165.36 | 10801.84 | 5.35 | 8 | 2 | 0.26 | 1.28 | 0.05 | 0.08 | 6 | 0.48 | 1.28 |
| | 100 | 7565.40 | 10801.91 | 10.76 | 3 | 7 | 0.66 | 1.30 | 0.36 | 0.69 | 4 | 0.60 | 1.30 |
| | 200 | 7573.87 | 10802.23 | 2388.00 | 3 | 7 | 0.58 | 2.56 | 2.29 | 4.17 | 4 | 0.26 | 0.69 |
| 5 | 20 | 1.76 | 2.06 | 1.76 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 3278.79 | 10801.97 | 5.38 | 7 | 3 | 0.73 | 2.50 | 0.11 | 0.13 | 5 | 1.33 | 3.45 |
| | 100 | 6565.34 | 10801.95 | 304.38 | 4 | 6 | 1.56 | 5.80 | 0.52 | 0.66 | 2 | 1.70 | 4.29 |
| | 200 | 7569.25 | 10802.22 | 33.14 | 3 | 7 | 0.58 | 1.53 | 4.03 | 8.83 | 2 | 0.72 | 1.53 |
| 10 | 50 | 423.92 | 2180.14 | 150.13 | 10 | 0 | 0.00 | 0.00 | 0.14 | 0.17 | 9 | 3.00 | 30.00 |
| | 100 | 2572.14 | 10801.25 | 491.99 | 9 | 1 | 0.50 | 5.00 | 0.76 | 0.98 | 7 | 1.40 | 5.00 |
| | 200 | 4450.37 | 10801.75 | 182.08 | 6 | 4 | 1.33 | 4.35 | 6.58 | 11.45 | 3 | 2.23 | 6.52 |
| 20 | 100 | 1.88 | 4.14 | 1.88 | 10 | 0 | 0.00 | 0.00 | 1.45 | 2.08 | 8 | 2.86 | 14.29 |
| | 200 | 3.43 | 4.35 | 3.43 | 10 | 0 | 0.00 | 0.00 | 8.35 | 16.27 | 7 | 2.00 | 6.67 |
| Average | | 3053.71 | 6369.43 | 231.16 | 139(sum) | 50(sum) | 0.37 | 1.47 | 1.42 | 2.89 | 122(sum) | 0.95 | 4.45 |

**Table 5.2:** Performance of the Solution Approaches for the Non-Resumable Jobs Case When $T_W$ is 50 and $T_U$ is 10.

| | | MILP | | | | | | | Heuristic-NR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non-optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 1.59 | 1.75 | 1.59 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.75 | 2.61 | 1.75 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 8641.94 | 10801.99 | 18.38 | 2 | 8 | 0.95 | 2.26 | 0.07 | 0.11 | 2 | 0.95 | 2.26 |
| | 100 | 10801.81 | 10801.94 | 255.83 | 0 | 8 | 1.32 | 2.37 | 0.81 | 1.19 | 0 | 0.72 | 1.20 |
| | 200 | 10802.05 | 10802.28 | 2017.60 | 0 | 4 | 1.28 | 1.82 | 6.63 | 13.47 | 0 | 0.38 | 0.67 |
| 3 | 10 | 1.09 | 1.36 | 1.09 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.75 | 4.72 | 1.75 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 9 | 0.19 | 1.89 |
| | 50 | 7411.40 | 10801.50 | 66.84 | 4 | 6 | 1.45 | 3.82 | 0.12 | 0.23 | 3 | 1.54 | 3.44 |
| | 100 | 10801.45 | 10801.53 | 1676.73 | 0 | 8 | 1.85 | 5.19 | 0.69 | 1.00 | 0 | 0.97 | 1.82 |
| | 200 | 10801.69 | 10801.91 | 870.16 | 0 | 9 | 1.47 | 2.44 | 4.38 | 7.42 | 0 | 0.50 | 1.02 |
| 5 | 20 | 3.22 | 8.45 | 3.22 | 10 | 0 | 0.00 | 0.00 | 0.02 | 0.03 | 9 | 0.31 | 3.08 |
| | 50 | 618.21 | 3657.55 | 429.59 | 10 | 0 | 0.00 | 0.00 | 0.13 | 0.20 | 5 | 0.47 | 1.95 |
| | 100 | 10801.72 | 10801.97 | 1467.95 | 0 | 10 | 3.39 | 7.91 | 1.15 | 1.97 | 0 | 2.75 | 5.59 |
| | 200 | 10801.92 | 10802.03 | 1353.87 | 0 | 10 | 1.91 | 3.94 | 6.72 | 8.27 | 0 | 1.43 | 2.24 |
| 10 | 50 | 574.62 | 5398.69 | 63.94 | 10 | 0 | 0.00 | 0.00 | 0.21 | 0.27 | 6 | 1.29 | 5.00 |
| | 100 | 10801.66 | 10801.84 | 947.85 | 0 | 10 | 6.02 | 11.04 | 1.60 | 2.13 | 0 | 6.15 | 11.04 |
| | 200 | 10801.83 | 10801.92 | 666.50 | 0 | 10 | 6.35 | 9.82 | 15.22 | 19.53 | 0 | 5.11 | 8.77 |
| 20 | 100 | 561.55 | 3438.72 | 337.49 | 10 | 0 | 0.00 | 0.00 | 4.48 | 5.78 | 4 | 3.41 | 8.70 |
| | 200 | 10801.38 | 10801.53 | 1365.05 | 0 | 10 | 3.80 | 6.67 | 29.15 | 37.23 | 0 | 5.23 | 7.14 |
| Average | | 6054.35 | 6912.33 | 607.74 | 86(sum) | 93(sum) | 1.57 | 3.02 | 3.76 | 5.20 | 68(sum) | 1.65 | 3.46 |

Another column in Tables 5.1 and 5.2 illustrates the number of non-optimum integer solutions obtained by the MILP. We do not have any information about the optimality of the problem, so obtained integer solution might be either optimum, or not. MILP could not find any integer solution in some problem instances such as $n$, $m$, $T_W$, and $T_U$ are 2, 200, 50, and 10, respectively. The number of optimum solutions obtained for this set of 10 problems instances is 0, and the number of non-optimal integer solutions is 4. In addition the total number of integer solutions (optimum and non-optimum) slightly worsens when the number of machines decreases and number of jobs increases.

Percent error of the solution approaches is also illustrated in Tables 5.1 and 5.2. It is obvious that the percent error for each solution approach is zero when the number of jobs is small (i.e., when $n$ is 10, or 20). For each solution approach, percent error increases when $n$ increases. The percent error of the solution approaches fluctuates randomly when the number of machines increases for the problem sets with $p_j \sim U[1, 10]$. On the contrary, the performance of each solution approach decreases as the number of machines increases for the problem sets with $p_j \sim$ U[1, 50].

To summarize the performance of solution approaches, it can be concluded that the performance of both MILP and the algorithm Heuristic-NR decrease as the problem size increases.

Table 5.3 illustrates the effects of changing maximum working time $T_W$ and unavailability time $T_U$ of the machines to the performance of solution approaches. To construct Table 5.3, we use the last row of Tables 5.1 and 5.2 and Tables B-1 to B-16, which provides the overall average of 190 problem instances. As can be seen from Table 5.3, $T_W$ and $T_U$ values have significant effect on the CPU time of the MILP. The CPU time of MILP decreases as $T_W$ increases. This is due to the fact that the number of working batches decreases when $T_W$ increases. However, increasing $T_U$ value increases the CPU time of MILP. The same observations can be seen in the average CPU time for the best integer solution. Another observation is that the CPU time of the MILP in the problem set with $p_j \sim U[1, 50]$ is higher than the problem sets with $p_j \sim U[1, 10]$.

As can be seen from Table 5.3, change in $T_W$ time does not show any consistent effect on the CPU time of the algorithm Heuristic-NR. On the contrary, increasing $T_U$ values has slightly negative effect on the CPU of the algorithm in the problem sets with $p_j \sim U[1, 10]$, and it has no significant effect for the problem sets with $p_j \sim U[1, 50]$. Thus, it can be concluded that the range of the processing times has a significant effect on the CPU time of the algorithm. The CPU time of the problem sets with $p_j \sim U[1, 10]$ is smaller than that of the problem sets with $p_j \sim U[1, 10]$.

In Table 5.3, the number of optimum solutions obtained by the solution approaches increases as the value of $T_W$ increases since the number of working batches is small in large maximum working times. However, $T_U$ values do not have any effect on the number of optimum solutions obtained by the solution approaches. The number of optimum solutions obtained in the problem sets with $p_j \sim U[1, 10]$ is higher than the problem set with $p_j \sim U[1, 50]$. This is due to the fact that the range of solution space to be searched for an integer solution is relatively large for problem sets with $p_j \sim U[1, 50]$. $T_W$ value also has a significant effect on the total number of integer solutions. It is seen that when $T_W$ is 50 and $T_U$ is 10, the number of integer solutions obtained is 179 (i.e., 86+93=179) out of 190 problem instances. Therefore, the MILP could not find any integer solution in 11 problem instances out of 190. On the contrary, the number of problem instances without integer solution becomes zero when $T_W$ is 150 and $T_U$ is 30. If $T_W$ is high and the range of the processing times is less, an integer solution can be found easily in the MILP.

It is obvious that $T_W$ and $T_U$ times have significant effect on the percent error of both MILP and heuristic algorithm. Performance of the solution approaches improves when $T_W$ increases due to the number of working batches. Increasing $T_U$ values reduce the performance of the solution approaches by increasing the percent error since the solution space to be searched increses with the increase in $T_U$.

**Table 5.3:** Performance of the Solution Approaches for the Non-Resumable Jobs Case When the $T_W$ and $T_U$ Changes

| | | | MILP | | | | | | | Heuristic-NR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $T_w$ | $T_U$ | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| $U$[1, 10] | 10 | 2 | 3053.71 | 6369.43 | 231.16 | 139 | 50 | 0.37 | 1.47 | 1.42 | 2.89 | 122 | 0.95 | 4.45 |
| | | 5 | 2929.65 | 6420.99 | 188.49 | 143 | 46 | 0.47 | 2.15 | 1.77 | 4.27 | 121 | 1.24 | 6.89 |
| | | 10 | 2633.63 | 5826.84 | 269.14 | 147 | 42 | 0.50 | 2.76 | 2.41 | 7.61 | 130 | 1.52 | 10.49 |
| | 20 | 4 | 830.41 | 3475.87 | 13.26 | 176 | 14 | 0.06 | 0.40 | 1.46 | 2.87 | 142 | 0.84 | 3.41 |
| | | 10 | 825.09 | 2922.91 | 14.36 | 176 | 14 | 0.14 | 1.10 | 1.86 | 4.35 | 143 | 0.88 | 4.26 |
| | | 20 | 944.98 | 2914.05 | 19.18 | 174 | 16 | 0.29 | 1.39 | 1.88 | 4.45 | 141 | 0.98 | 5.31 |
| | 30 | 6 | 404.54 | 2298.24 | 5.40 | 183 | 7 | 0.02 | 0.14 | 1.37 | 2.63 | 147 | 0.82 | 3.97 |
| | | 15 | 517.10 | 2844.93 | 2.91 | 181 | 9 | 0.05 | 0.32 | 1.47 | 3.04 | 147 | 0.87 | 4.48 |
| | | 30 | 579.89 | 2322.83 | 8.72 | 180 | 10 | 0.19 | 1.79 | 1.47 | 3.04 | 145 | 0.86 | 5.04 |
| $U$[1, 50] | 50 | 10 | 6054.35 | 6912.33 | 607.74 | 86 | 93 | 1.57 | 3.02 | 3.76 | 5.20 | 68 | 1.65 | 3.46 |
| | | 25 | 5784.14 | 7379.53 | 473.30 | 90 | 91 | 1.95 | 4.69 | 4.36 | 6.31 | 70 | 1.79 | 4.70 |
| | | 50 | 6235.78 | 7646.24 | 809.64 | 83 | 99 | 3.57 | 12.78 | 5.45 | 11.15 | 67 | 2.17 | 6.71 |
| | 100 | 20 | 3612.49 | 5438.93 | 290.02 | 128 | 61 | 0.46 | 1.20 | 3.34 | 4.87 | 85 | 1.04 | 2.80 |
| | | 50 | 3588.54 | 4836.67 | 318.69 | 128 | 61 | 0.52 | 1.75 | 3.59 | 6.29 | 86 | 0.98 | 2.71 |
| | | 100 | 3673.17 | 4789.80 | 295.42 | 127 | 62 | 0.47 | 1.73 | 3.59 | 6.26 | 84 | 0.95 | 2.81 |
| | 150 | 30 | 2696.81 | 4055.94 | 214.97 | 144 | 46 | 0.53 | 1.66 | 3.45 | 5.27 | 90 | 1.18 | 3.31 |
| | | 75 | 2973.29 | 4167.07 | 301.97 | 140 | 50 | 0.99 | 3.33 | 3.63 | 6.02 | 89 | 1.61 | 4.80 |
| | | 150 | 2917.58 | 4884.92 | 310.06 | 142 | 48 | 2.28 | 6.43 | 3.62 | 6.02 | 87 | 2.32 | 7.34 |
| | Average | | 2791.95 | 4750.42 | 243.02 | 143 | 46 | 0.80 | 2.67 | 2.77 | 5.14 | 109 | 1.26 | 4.83 |

The solution approaches provide better solutions to the problems with a small range of processing times since the difference between non-integer and best-integer solutions is small and thus the solution space to be searched for an integer solution is reduced. Morover, the number of alternative solutions obtained increases as the range of the processing times decreases since the probability of having equal processing times for a job on different machines increases as the range ot processing times decreases.

As a summary, the average CPU time of the algorithm Heuristic-NR is 1000 times smaller than the average CPU time of the MILP. The number of optimum solutions obtained in the MILP is more. On the other hand, the MILP cannot find any integer solution for some problem instances although the algorithm Heuristic-NR always finds an integer solution. Although the percent error of the algorithm Heuristic-NR is smaller in some problem sets such as $T_W$ is 50 and $T_U$ is 50, the percent error of the MILP is generally less than the percent error of the algorithm Heuristic-NR. The average percent error of MILP and the algorithm Heuristic-NR is 0.80, and 1.26, respectively. For the problem instances with known optimal solutions, the percent error of the algorithm Heuristic-NR is between 0.50 and 1.26. Therefore, we can conclude that the algorithm Heuristic-NR is notably effective since it performs very well in short average CPU times with 2.77 seconds.

### 5.3.1.2 Effect of the phases in the heuristic algorithm

As it is mentioned in Chapter 4, the heuristic algorithm has four phases, and the application of each phase decreases the percent error of the heuristic's makespan from the optimal one but increases the computational effort. Thus, a user may not want to apply some of the phases in order to get a solution in very short time. Therefore, we discuss the performance of the each phase in this section.

We observe that the number of jobs and the number of machines nearly show same effects when the maximum working time and unavailability time are fixed. Therefore, we only report Table 5.4 in this section to show the effects of $n$ and $m$ in problems sets with $p_j \sim U[1, 10]$ when $T_W$ is 10 and $T_U$ is 2. For the problem sets with $p_j \sim U[1,$

50], Table 5.5 is only reported in this section which shows the effects of $n$ and $m$ when $T_W$ is 50 and $T_U$ is 10.

**Table 5.4:** Effects of Phases in the Algorithm Heuristic-NR for the Non-Resumable Jobs Case When $T_W$ is 10 and $T_U$ is 2

| | | Phase 1 | Phase 2 | | Phase 3 | | Phase 4 | |
|---|---|---|---|---|---|---|---|---|
| $m$ | $n$ | Average CPU time | Average CPU time | Average Percent Improvement | Average CPU time | Average Percent Improvement | Average CPU time | Average Percent Improvement |
| 2 | 10 | 0.01 | 0.01 | 9.66 | 0.01 | 0.43 | 0.01 | 0.00 |
| | 20 | 0.00 | 0.00 | 7.74 | 0.00 | 0.00 | 0.00 | 0.00 |
| | 50 | 0.00 | 0.00 | 4.43 | 0.01 | 0.30 | 0.04 | 0.00 |
| | 100 | 0.00 | 0.00 | 8.63 | 0.01 | 0.04 | 0.39 | 0.31 |
| | 200 | 0.00 | 0.02 | 3.68 | 0.03 | 0.05 | 2.07 | 0.11 |
| 3 | 10 | 0.00 | 0.00 | 9.81 | 0.00 | 4.60 | 0.00 | 0.71 |
| | 20 | 0.00 | 0.00 | 12.96 | 0.00 | 1.37 | 0.01 | 0.77 |
| | 50 | 0.00 | 0.00 | 9.43 | 0.00 | 0.47 | 0.05 | 0.38 |
| | 100 | 0.00 | 0.01 | 8.59 | 0.01 | 0.31 | 0.36 | 0.44 |
| | 200 | 0.00 | 0.01 | 5.22 | 0.03 | 0.00 | 2.29 | 0.13 |
| 5 | 20 | 0.00 | 0.00 | 18.41 | 0.00 | 0.00 | 0.01 | 2.91 |
| | 50 | 0.00 | 0.00 | 23.49 | 0.01 | 1.22 | 0.11 | 1.58 |
| | 100 | 0.01 | 0.01 | 16.41 | 0.02 | 0.83 | 0.52 | 0.98 |
| | 200 | 0.01 | 0.03 | 9.56 | 0.03 | 0.34 | 4.03 | 0.43 |
| 10 | 50 | 0.00 | 0.00 | 25.38 | 0.01 | 1.00 | 0.14 | 6.76 |
| | 100 | 0.00 | 0.01 | 19.55 | 0.02 | 0.00 | 0.76 | 4.51 |
| | 200 | 0.01 | 0.02 | 14.28 | 0.04 | 0.20 | 6.58 | 0.88 |
| 20 | 100 | 0.01 | 0.01 | 31.39 | 0.03 | 1.11 | 1.45 | 3.75 |
| | 200 | 0.01 | 0.03 | 17.56 | 0.07 | 0.00 | 8.35 | 0.00 |
| Average | | 0.00 | 0.01 | 13.48 | 0.02 | 0.65 | 1.43 | 1.30 |

**Table 5.5:** Effects of Phases in the Algorithm Heuristic-NR for the Non-Resumable Jobs Case When $T_W$ is 50 and $T_U$ is 10

| | | Phase 1 | Phase 2 | | Phase 3 | | Phase 4 | |
|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Average CPU time | Average Percent Improvement | Average CPU time | Average Percent Improvement | Average CPU time | Average Percent Improvement |
| 2 | 10 | 0.00 | 0.00 | 11.34 | 0.00 | 1.15 | 0.00 | 0.00 |
| | 20 | 0.00 | 0.00 | 10.85 | 0.00 | 1.28 | 0.00 | 0.69 |
| | 50 | 0.00 | 0.00 | 9.21 | 0.00 | 0.18 | 0.07 | 0.33 |
| | 100 | 0.00 | 0.01 | 7.46 | 0.02 | 0.25 | 0.81 | 0.21 |
| | 200 | 0.01 | 0.03 | 5.36 | 0.04 | 0.03 | 6.63 | 0.14 |
| 3 | 10 | 0.00 | 0.00 | 18.39 | 0.00 | 0.00 | 0.00 | 0.57 |
| | 20 | 0.00 | 0.00 | 11.65 | 0.00 | 0.72 | 0.01 | 0.58 |
| | 50 | 0.00 | 0.00 | 14.46 | 0.00 | 1.40 | 0.12 | 0.46 |
| | 100 | 0.00 | 0.01 | 7.62 | 0.02 | 0.50 | 0.69 | 0.28 |
| | 200 | 0.00 | 0.02 | 6.57 | 0.03 | 0.10 | 4.38 | 0.19 |
| 5 | 20 | 0.00 | 0.00 | 33.98 | 0.00 | 0.25 | 0.02 | 4.15 |
| | 50 | 0.00 | 0.00 | 28.94 | 0.01 | 0.60 | 0.13 | 1.32 |
| | 100 | 0.00 | 0.01 | 20.20 | 0.01 | 0.23 | 1.15 | 0.88 |
| | 200 | 0.00 | 0.02 | 17.56 | 0.03 | 0.14 | 6.72 | 0.46 |
| 10 | 50 | 0.00 | 0.00 | 44.46 | 0.01 | 0.42 | 0.21 | 4.49 |
| | 100 | 0.00 | 0.01 | 31.06 | 0.03 | 1.30 | 1.60 | 1.65 |
| | 200 | 0.01 | 0.02 | 23.68 | 0.06 | 0.54 | 15.22 | 1.09 |
| 20 | 100 | 0.01 | 0.02 | 39.97 | 0.04 | 3.87 | 4.48 | 7.06 |
| | 200 | 0.02 | 0.04 | 43.42 | 0.14 | 0.00 | 29.15 | 2.41 |
| Average | | 0.00 | 0.01 | 20.33 | 0.02 | 0.68 | 3.76 | 1.42 |

As can be seen from Tables 5.4 and 5.5, the CPU times for Phases 1, 2, and 3 of the algorithm Heuristic-NR are close to zero, so they can be negligible. Only Phase 4 requires notably large CPU time for heuristic algorithm. The number of jobs and machines are highly significant for the CPU time for Phase 4. Increase in the value of both parameters increases the CPU time.

Phase 2 of the algorithm Heuristic-NR is the most effective phase. The average improvement of Phase 2 reduces as the number of jobs increases, and increases as the number of machines increases. Improvement of Phase 3 is less than the other two phases. The number of jobs and machines has no significant effect on the percent

improvement of Phase 3. It is due to the fact that the percent improvement of Phase 3 shows random pattern when the number of jobs and machines change. Phase 4 provides more improvement than Phase 3 and it is more effective when the number of jobs is small. Also, it shows little increase as the number of machines increases.

Table 5.6 is preapared to investigate the performance of phases in the algorithm Heuristic-NR with respect to the changes in $T_W$ and $T_U$ values. The CPU time for Phases 1, 2 and 3 is too small to make comment about the effects of $T_W$ and $T_U$ times. The CPU time for Phase 4 is relatively high and it does not illustrate any relation with different $T_W$ and $T_U$ values. In other words, it shows random pattern when $T_W$ and $T_U$ values change. The average CPU time of Phase 4 in the problem sets with $p_j \sim U[1, 50]$ is higher than the one in the sets with $p_j \sim U[1, 10]$.

**Table 5.6:** Performance of Phases in the Algorithm Heuristic-NR for the Non-Resumable Jobs Case When the $T_W$ and $T_U$ Changes

| | | | Phase 1 | Phase 2 | | Phase 3 | | Phase 4 | |
|---|---|---|---|---|---|---|---|---|---|
| | $T_W$ | $T_U$ | Average CPU time | Average CPU time | Average Percent Improvement | Average CPU time | Average Percent Improvement | Average CPU time | Average Percent Improvement |
| $U[1, 10]$ | 10 | 2 | 0.00 | 0.01 | 13.48 | 0.02 | 0.65 | 1.43 | 1.30 |
| | | 5 | 0.00 | 0.05 | 13.79 | 0.02 | 0.58 | 1.77 | 1.52 |
| | | 10 | 0.00 | 0.01 | 14.01 | 0.02 | 0.51 | 2.41 | 1.75 |
| | 20 | 4 | 0.00 | 0.01 | 13.80 | 0.02 | 0.66 | 1.46 | 1.23 |
| | | 10 | 0.00 | 0.01 | 14.20 | 0.02 | 0.67 | 1.86 | 1.49 |
| | | 20 | 0.00 | 0.01 | 14.69 | 0.02 | 0.67 | 1.88 | 1.73 |
| | 30 | 6 | 0.00 | 0.01 | 13.48 | 0.02 | 0.74 | 1.37 | 1.17 |
| | | 15 | 0.00 | 0.01 | 13.69 | 0.02 | 0.76 | 1.47 | 1.34 |
| | | 30 | 0.00 | 0.01 | 13.79 | 0.02 | 0.78 | 1.47 | 1.48 |
| $U[1, 50]$ | 50 | 10 | 0.00 | 0.01 | 20.33 | 0.02 | 0.68 | 3.76 | 1.42 |
| | | 25 | 0.00 | 0.01 | 21.59 | 0.02 | 0.73 | 4.36 | 1.48 |
| | | 50 | 0.00 | 0.01 | 22.90 | 0.03 | 0.71 | 5.45 | 1.47 |
| | 100 | 20 | 0.00 | 0.01 | 19.72 | 0.02 | 0.77 | 3.34 | 1.39 |
| | | 50 | 0.00 | 0.01 | 20.35 | 0.02 | 0.81 | 3.59 | 1.43 |
| | | 100 | 0.00 | 0.01 | 20.97 | 0.02 | 0.84 | 3.59 | 1.45 |
| | 150 | 30 | 0.00 | 0.01 | 20.07 | 0.02 | 0.70 | 3.45 | 1.43 |
| | | 75 | 0.00 | 0.01 | 21.13 | 0.02 | 0.68 | 3.63 | 1.48 |
| | | 150 | 0.00 | 0.01 | 22.23 | 0.02 | 0.64 | 3.62 | 1.53 |
| Average | | | 0.00 | 0.01 | 17.46 | 0.02 | 0.70 | 2.77 | 1.45 |

As can be seen in Table 5.6, the percent improvement on the makespan by Phase 2 strongly increases as the maximum working time and the range of processing times increases. The unavailablity time is significantly effective and improves performance of Phase 2.

Phase 3 has a slight improvement when the maximum working time increases for the problem sets with $p_j \sim U[1, 10]$, and it fluctuates randomly in problem sets with $p_j \sim U[1, 50]$. Furthermore, Phase 3 does not illustrate any noticeable behavior when the unavailability time increases. In the problem sets with $p_j \sim U[1, 50]$, the percent improvement of Phase 3 is better as compared to the problem sets with $p_j \sim U[1, 10]$.

From the Table 5.6, it is observed that percent improvement obtained by Phase 4 increases as unavailability time increases. The percent improvement of Phase 4 decreases as the maximum working time increases in the problem sets with $p_j \sim U[1, 10]$. However, it does not show any systematic behavior for percent improvement of Phase 4 in the problem sets with $p_j \sim U[1, 50]$. It is observed that the effect of Phase 4 in problem set with $p_j \sim U[1, 50]$ is similar to the problem set with $p_j \sim U[1, 50]$. Therefore, the range of the processing times does not affect the performance of Phase 4.

Effects of the phases in the algorithm Heuristic-NR are illustrated in Figure 5.1. It is obvious that the greatest average improvement is achieved by Phase 2, and its average improvement is 17.46 percent. In the heuristic algorithm, the average improvement obtained by Phase 4 is better than the one obtained by Phase 3 since the percent improvements by Phases 4 and 3 are 1.45 and 0.70, respectively.

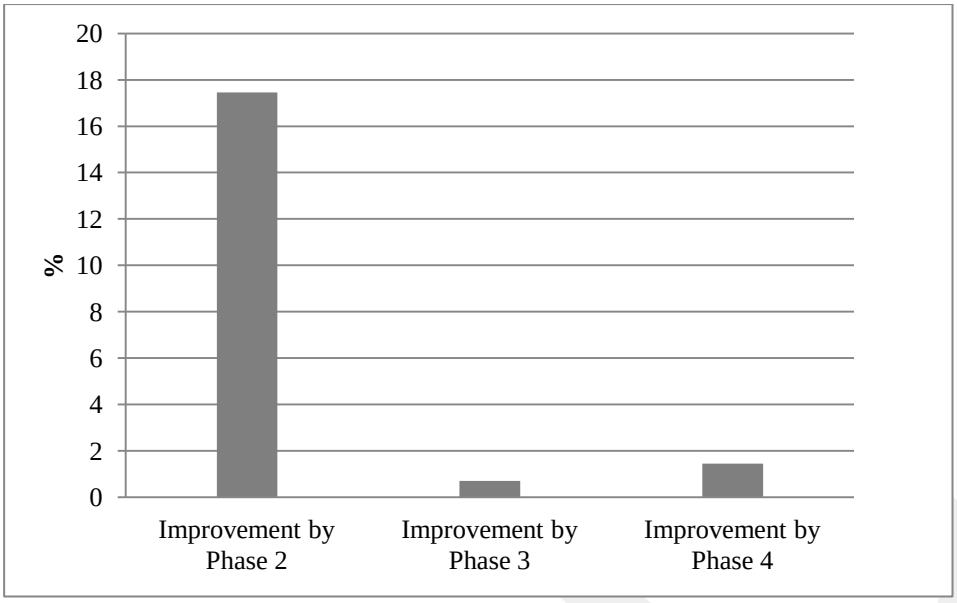


**Figure 5.1:** Percent Improvement on the Makespan Obtained by the Phases of the Algorithm Heuristic-NR for the Non-Resumable Jobs Case

From Table 5.6, it is concluded that the first three phases requires very small CPU times. The average CPU time is less than 0.02 second for each phase. Phase 4 requires more CPU time and its average CPU time is about 2.77 seconds. Thus, we can propose the user either to employ the first three phases with very short CPU times or to use all phases to find better solutions by consuming more time which is less than 3 seconds.

### 5.3.2 Case 2: Resumable jobs

We will discuss performance of solution methods, and the effects of the phases in heuristic algorithm for resumable jobs case in this part of the chapter.

### 5.3.2.1 Performance of the solution approaches

Firstly, we examine the effects of number of jobs $n$ and number of machines $m$ to the performance of solution approaches when the maximum working time $T_W$ and unavailability time $T_U$ of the machines are fixed. Then, the effects of $T_W$ and $T_U$ times to the performance of solution approaches will be discussed.

For each $T_W$ and $T_U$ combination, we observe that change in the number of jobs and number of machines show approximately same characteristics on the performance of the solution approaches. Therefore, we report Table 5.7 to investigate the effects of changes in the number of jobs and machines for problem set with $p_j \sim U[1, 10]$ when $T_W$ is 10 and $T_U$ is 2. The effects of change in the number of jobs and machines for other $T_W$ and $T_U$ combinations in problem set with $p_j \sim U[1, 10]$ can be observed in Tables C-1 to C-8 in the appendix. Moreover, Table 5.8 show the effects of change in the number of jobs and machines when the $T_W$ is 50 and $T_U$ is 10 for problem set with $p_j \sim U[1, 50]$. For the other $T_W$ and $T_U$ combinations for problem sets with $p_j \sim U[1, 50]$, we can see the effects of change in the number of jobs and machines on the performance of the solution approaches in Tables C-9 to C-16 in the appendix. From tables 5.7, 5.8 C-1 to C-16, we observe that the proposed heuristic algorithm Heuristic-R with Initial Schedules 1, 2, 3 and 4 find the best solution for 2820, 2202, 2293, and 2273 problem instances, respectively. From these results, it is clear that the performance of the proposed heuristic with Initial Schedule 1 is better than the others.

Tables 5.7 and 5.8 indicate that increasing the number of jobs increases the maximum and average CPU times of the each solution approaches. The CPU time to find the best integer solution is approximately equal to the CPU of small sized problems, and it is smaller than the CPU time of the large sized problems. The CPU time of the solution approaches also increases as the number of machines increases.

**Table 5.7:** Performance of the Solution Approaches for the Resumable Jobs Case When $T_W$ is 10 and $T_U$ is 2

| | | MILP | | | | | | | Heuristic-R | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 3.10 | 3.30 | 3.10 | 10 | 0 | 0.00 | 0.00 | 2.72 | 3.47 | 10 | 0.00 | 0.00 |
| | 20 | 3.17 | 3.26 | 3.17 | 10 | 0 | 0.00 | 0.00 | 2.81 | 3.35 | 10 | 0.00 | 0.00 |
| | 50 | 3.19 | 3.37 | 3.19 | 10 | 0 | 0.00 | 0.00 | 2.80 | 3.36 | 10 | 0.00 | 0.00 |
| | 100 | 3.18 | 3.34 | 3.18 | 10 | 0 | 0.00 | 0.00 | 2.94 | 3.30 | 10 | 0.00 | 0.00 |
| | 200 | 3.35 | 3.89 | 3.35 | 10 | 0 | 0.00 | 0.00 | 3.55 | 4.76 | 10 | 0.00 | 0.00 |
| 3 | 10 | 2.94 | 3.05 | 2.94 | 10 | 0 | 0.00 | 0.00 | 2.67 | 3.21 | 10 | 0.00 | 0.00 |
| | 20 | 3.11 | 3.20 | 3.11 | 10 | 0 | 0.00 | 0.00 | 2.72 | 2.98 | 10 | 0.00 | 0.00 |
| | 50 | 3.21 | 3.35 | 3.21 | 10 | 0 | 0.00 | 0.00 | 2.84 | 3.14 | 9 | 0.13 | 1.27 |
| | 100 | 3.29 | 3.48 | 3.29 | 10 | 0 | 0.00 | 0.00 | 3.12 | 3.84 | 8 | 0.13 | 0.63 |
| | 200 | 42.05 | 348.95 | 4.65 | 10 | 0 | 0.00 | 0.00 | 4.66 | 6.31 | 9 | 0.03 | 0.34 |
| 5 | 20 | 3.14 | 3.35 | 3.14 | 10 | 0 | 0.00 | 0.00 | 2.75 | 3.27 | 10 | 0.00 | 0.00 |
| | 50 | 3.77 | 5.13 | 3.77 | 10 | 0 | 0.00 | 0.00 | 2.89 | 3.54 | 6 | 1.09 | 3.45 |
| | 100 | 66.85 | 264.69 | 3.73 | 10 | 0 | 0.00 | 0.00 | 3.29 | 3.80 | 8 | 0.29 | 1.56 |
| | 200 | 3341.50 | 10804.09 | 3.83 | 7 | 3 | 0.14 | 0.61 | 6.48 | 9.15 | 4 | 0.36 | 0.75 |
| 10 | 50 | 834.85 | 6505.22 | 4.01 | 10 | 0 | 0.00 | 0.00 | 2.92 | 3.49 | 9 | 3.00 | 30.00 |
| | 100 | 2890.96 | 10803.61 | 261.18 | 8 | 2 | 0.78 | 4.00 | 3.74 | 4.83 | 6 | 1.68 | 5.00 |
| | 200 | 5403.75 | 10803.73 | 3.69 | 5 | 5 | 0.48 | 1.46 | 11.05 | 18.81 | 2 | 1.40 | 2.50 |
| 20 | 100 | 5696.64 | 10803.45 | 3.73 | 6 | 4 | 2.22 | 9.28 | 5.03 | 7.58 | 6 | 2.22 | 9.28 |
| | 200 | 9723.58 | 10803.76 | 3.67 | 2 | 8 | 2.24 | 5.26 | 22.03 | 39.03 | 0 | 4.24 | 6.89 |
| Average | | 1475.56 | 3219.80 | 17.05 | 168(sum) | 22(sum) | 0.31 | 1.08 | 4.79 | 6.91 | 147 | 0.77 | 3.25 |

**Table 5.8:** Performance of the Solution Approaches for the Resumable Jobs Case When $T_W$ is 50 and $T_U$ is 10

| | | MILP | | | | | | | Heuristic-R | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 3.26 | 3.42 | 3.26 | 10 | 0 | 0.00 | 0.00 | 3.21 | 3.54 | 10 | 0.00 | 0.00 |
| | 20 | 3.20 | 3.43 | 3.20 | 10 | 0 | 0.00 | 0.00 | 3.25 | 3.60 | 10 | 0.00 | 0.00 |
| | 50 | 3.24 | 3.34 | 3.24 | 10 | 0 | 0.00 | 0.00 | 3.14 | 3.64 | 10 | 0.00 | 0.00 |
| | 100 | 3.28 | 3.44 | 3.28 | 10 | 0 | 0.00 | 0.00 | 3.43 | 4.10 | 8 | 0.02 | 0.16 |
| | 200 | 3.31 | 3.46 | 3.31 | 10 | 0 | 0.00 | 0.00 | 4.64 | 5.56 | 4 | 0.03 | 0.08 |
| 3 | 10 | 3.16 | 3.26 | 3.16 | 10 | 0 | 0.00 | 0.00 | 3.31 | 3.86 | 10 | 0.00 | 0.00 |
| | 20 | 3.25 | 3.37 | 3.25 | 10 | 0 | 0.00 | 0.00 | 3.26 | 3.81 | 10 | 0.00 | 0.00 |
| | 50 | 3.25 | 3.41 | 3.25 | 10 | 0 | 0.00 | 0.00 | 3.29 | 3.72 | 7 | 0.11 | 0.52 |
| | 100 | 3.28 | 3.44 | 3.28 | 10 | 0 | 0.00 | 0.00 | 3.50 | 3.95 | 8 | 0.13 | 0.96 |
| | 200 | 4.44 | 12.32 | 4.40 | 10 | 0 | 0.00 | 0.00 | 6.24 | 7.26 | 1 | 0.18 | 0.88 |
| 5 | 20 | 3.15 | 3.50 | 3.15 | 10 | 0 | 0.00 | 0.00 | 3.15 | 4.41 | 8 | 0.47 | 3.08 |
| | 50 | 3.57 | 4.75 | 3.57 | 10 | 0 | 0.00 | 0.00 | 3.30 | 3.48 | 7 | 0.33 | 1.95 |
| | 100 | 5.07 | 8.67 | 5.07 | 10 | 0 | 0.00 | 0.00 | 4.50 | 5.19 | 2 | 0.40 | 0.65 |
| | 200 | 161.77 | 1433.51 | 11.80 | 10 | 0 | 0.00 | 0.00 | 10.37 | 14.00 | 1 | 0.32 | 0.89 |
| 10 | 50 | 50.63 | 467.36 | 4.23 | 10 | 0 | 0.00 | 0.00 | 3.55 | 4.19 | 6 | 1.13 | 5.00 |
| | 100 | 695.85 | 3671.28 | 575.79 | 10 | 0 | 0.00 | 0.00 | 6.58 | 7.93 | 0 | 1.52 | 2.70 |
| | 200 | 7296.90 | 10804.56 | 127.80 | 5 | 5 | 0.44 | 1.33 | 23.71 | 39.70 | 0 | 2.04 | 7.10 |
| 20 | 100 | 1218.39 | 3445.26 | 328.75 | 10 | 0 | 0.00 | 0.00 | 10.59 | 11.82 | 6 | 1.71 | 4.55 |
| | 200 | 10803.68 | 10803.96 | 11.48 | 0 | 10 | 3.13 | 3.87 | 48.28 | 58.94 | 0 | 3.87 | 6.17 |
| Average | | 1066.98 | 1615.04 | 58.17 | 175(sum) | 15 | 0.19 | 0.27 | 7.96 | 10.14 | 108(sum) | 0.65 | 1.83 |

When the number of machines is large, increasing the number of jobs reduces the number of optimum solutions obtained. The number of optimum solutions obtained also decreases as the number of machines increases. Therefore, the performance of the solution approaches worsens in the large sized problems.

In small sized problems, solution approaches provide better results and their percent error is close to zero. Increasing the number of jobs and machines increases the percent error of each solution approach. Therefore, we can conclude that the performance of solution approaches worsens when the problem size increases.

Effects of changing the maximum working time $T_W$ and the unavailability time $T_U$ of the machines on the performance of solution approaches are illustrated in Table 5.9. Table 5.9 shows that $T_W$ and $T_U$ values have no significant effect on the CPU time of the MILP and the algorithm Heuristic-R. Table 5.9 indicates that the CPU time of the MILP in the problem set with $p_j \sim U[1, 10]$ is higher than the problem set with $p_j \sim U[1, 50]$, The CPU time of the algorithm Heuristic-R are small in the problem set with $p_j \sim U[1, 10]$.

As can be seen from Table 5.9, the number of optimum solutions obtained by each solution approach is not affected by the maximum working time $T_W$ and the unavailability time $T_U$. For each solution approach, the range of the processing times has a significant effect on the number of optimum solutions obtained. We observe that the number of optimum solutions obtained in the problem set with $p_j \sim U[1, 10]$ is higher than the problem set with $p_j \sim U[1, 50]$. We also observe that each approach find an integer solution for each problem.

**Table 5.9:** Performance of the Solution Approaches for the Resumable Jobs Case When the $T_W$ and $T_U$ Changes

| | | | MILP | | | | | | | Heuristic-R | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $T_w$ | $T_U$ | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| $U$[1, 10] | 10 | 2 | 1475.56 | 3219.80 | 17.05 | 168 | 22 | 0.31 | 1.08 | 4.79 | 6.91 | 147 | 0.77 | 3.25 |
| | | 5 | 1366.91 | 2975.03 | 3.43 | 169 | 21 | 0.30 | 1.04 | 6.89 | 9.75 | 143 | 0.88 | 4.58 |
| | | 10 | 1401.10 | 2955.66 | 8.07 | 168 | 22 | 0.26 | 0.91 | 7.49 | 11.22 | 142 | 1.05 | 7.03 |
| | 20 | 4 | 1388.15 | 2920.90 | 13.94 | 168 | 22 | 0.34 | 1.17 | 5.16 | 6.98 | 148 | 0.56 | 2.18 |
| | | 10 | 1529.70 | 2913.38 | 56.74 | 167 | 23 | 0.35 | 1.18 | 5.50 | 8.24 | 143 | 0.59 | 2.14 |
| | | 20 | 1369.20 | 2914.60 | 4.01 | 169 | 21 | 0.36 | 1.18 | 5.72 | 9.01 | 145 | 0.59 | 2.11 |
| | 30 | 6 | 1342.49 | 2894.03 | 3.98 | 169 | 21 | 0.35 | 1.17 | 4.54 | 6.32 | 144 | 0.68 | 2.45 |
| | | 15 | 1342.49 | 2906.93 | 3.98 | 166 | 24 | 0.36 | 1.14 | 5.88 | 8.45 | 141 | 0.77 | 3.20 |
| | | 30 | 1431.89 | 2896.25 | 3.60 | 167 | 23 | 0.35 | 1.12 | 6.01 | 8.10 | 146 | 0.68 | 3.66 |
| $U$[1, 50] | 50 | 10 | 1066.98 | 1615.04 | 58.17 | 175 | 15 | 0.19 | 0.27 | 7.96 | 10.14 | 108 | 0.65 | 1.83 |
| | | 25 | 1053.57 | 1588.42 | 137.49 | 176 | 14 | 0.19 | 0.27 | 9.56 | 15.53 | 105 | 0.64 | 1.94 |
| | | 50 | 1023.61 | 1741.35 | 28.83 | 175 | 15 | 0.20 | 0.47 | 9.85 | 15.30 | 102 | 0.55 | 1.10 |
| | 100 | 20 | 1131.25 | 1397.89 | 66.49 | 173 | 17 | 0.20 | 0.27 | 9.28 | 13.96 | 109 | 0.62 | 1.59 |
| | | 50 | 959.51 | 1397.50 | 65.98 | 177 | 13 | 0.18 | 0.26 | 9.60 | 14.53 | 105 | 0.61 | 1.50 |
| | | 100 | 1029.05 | 1397.29 | 90.62 | 176 | 14 | 0.18 | 0.26 | 9.86 | 14.52 | 108 | 0.51 | 1.29 |
| | 150 | 30 | 1017.43 | 1493.91 | 49.79 | 176 | 14 | 0.18 | 0.24 | 10.46 | 14.60 | 111 | 0.59 | 1.47 |
| | | 75 | 1041.18 | 1302.25 | 106.63 | 176 | 14 | 0.20 | 0.37 | 9.33 | 14.64 | 110 | 0.53 | 1.44 |
| | | 150 | 972.31 | 1295.94 | 48.53 | 176 | 14 | 0.20 | 0.39 | 9.34 | 13.76 | 108 | 0.52 | 1.38 |
| Average | | | 1219.02 | 2212.57 | 42.63 | 171.72 | 18.28 | 0.26 | 0.71 | 7.62 | 11.22 | 125.83 | 0.65 | 2.45 |

Percent error of the MILP and the algorithm Heuristic-R shows random pattern as the $T_W$ time changes. Therefore, $T_W$ has no effect on the performance of solution approaches. Morever, $T_U$ time has no significant effect on the performans of solution approaches. Table 5.9 indicates that the percent error of the solution approaches in the problem set with $p_j \sim U[1, 10]$ is worse than the problem set with $p_j \sim U[1, 50]$.

To summarize we can conclude that the MILP model is better than the algorithm Heuristic-R in the performance criteria such as the number of optimum solutions obtained and the percent error of the solution approaches. However, the CPU time of the MILP model is very high. Although the percent error of the algorithm Heuristic-R is worse than the MILP, the algorithm Heuristic-R solves the problems in very short time. Table 5.9 shows that average percent deviation of the Heurisitic-R from the makespan of the non-integer solution obtained by the MILP is 0.65 percent, and its average CPU time is 7.62 seconds. Therefore, we can conclude that the maximum deviation of Heuristic-R from the optimum solution is 0.65 percent.

### 5.3.2.2 Effect of the phases in the heuristic algorithm

We also examine the effects of phases in the algorithm Heuristic-R which is proposed for resumable jobs case. As in the case of non-resumable jobs, the number of jobs and the number of machines nearly show same effects when the maximum working time and unavailability time are constant. The effects of $n$ and $m$ for problem sets with $p_j \sim U[1, 10]$ when $T_W$ is 10 and $T_U$ is 2 are shown in Table 5.10. When $T_W$ is 50 and $T_U$ is 10, Table 5.11 is reported to illustrate the effects of $n$ and $m$ on the performance of phases for the problem sets with $p_j \sim U[1, 50]$.

Tables 5.10 and 5.11 show that increasing the number of jobs and machines slightly increases the CPU time for each phase. Tables 5.10 and 5.11 also indicate that the average improvement of each phase in the algorithm Heuristic-R decreases as the number of jobs increases. Moreover, the number of machines is also effective on the performance of the phases. Increasing the number of machines produces better performance for each phase in each heuristic.

**Table 5.10:** Effects of Phases in the Algorithm Heuristic-R for the Resumable Jobs Case When $T_W$ is 10 and $T_U$ is 2

| | | Phase 1 | Phase 2 | | Phase 3 | | Phase 4 | |
|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Average CPU time | Average Percent Improvement | Average CPU time | Average Percent Improvement | Average CPU time | Average Percent Improvement |
| 2 | 10 | 2.72 | 2.72 | 1.50 | 2.72 | 0.45 | 2.72 | 0.00 |
| | 20 | 2.81 | 2.81 | 1.84 | 2.81 | 0.00 | 2.81 | 0.00 |
| | 50 | 2.80 | 2.80 | 0.74 | 2.80 | 0.08 | 2.80 | 0.23 |
| | 100 | 2.78 | 2.78 | 0.55 | 2.79 | 0.07 | 2.94 | 0.07 |
| | 200 | 2.82 | 2.83 | 0.31 | 2.84 | 0.04 | 3.55 | 0.07 |
| 3 | 10 | 2.67 | 2.67 | 3.99 | 2.67 | 3.93 | 2.67 | 0.48 |
| | 20 | 2.72 | 2.72 | 3.82 | 2.72 | 0.70 | 2.72 | 0.00 |
| | 50 | 2.80 | 2.80 | 2.38 | 2.80 | 0.35 | 2.84 | 0.12 |
| | 100 | 2.82 | 2.83 | 1.48 | 2.84 | 0.37 | 3.12 | 0.38 |
| | 200 | 2.83 | 2.84 | 1.02 | 2.85 | 0.00 | 4.66 | 0.10 |
| 5 | 20 | 2.75 | 2.75 | 10.63 | 2.75 | 0.00 | 2.75 | 1.34 |
| | 50 | 2.80 | 2.80 | 4.10 | 2.80 | 0.51 | 2.89 | 1.81 |
| | 100 | 2.79 | 2.79 | 3.35 | 2.80 | 0.42 | 3.29 | 0.28 |
| | 200 | 2.97 | 2.97 | 1.87 | 2.99 | 0.07 | 6.48 | 0.28 |
| 10 | 50 | 2.72 | 2.72 | 9.50 | 2.73 | 1.71 | 2.92 | 4.62 |
| | 100 | 2.79 | 2.79 | 10.57 | 2.81 | 0.45 | 3.74 | 1.68 |
| | 200 | 3.03 | 3.04 | 5.92 | 3.08 | 0.00 | 11.05 | 0.65 |
| 20 | 100 | 3.02 | 3.02 | 9.92 | 3.05 | 0.00 | 5.03 | 6.25 |
| | 200 | 3.28 | 3.29 | 11.25 | 3.41 | 0.00 | 22.03 | 0.00 |
| Average | | 2.84 | 2.84 | 4.46 | 2.85 | 0.48 | 4.79 | 0.97 |

Tables 5.12 is prepared to investigate the effect of changing working time $T_W$ and unavailability time $T_U$ on the performance of the phases in the algorithm Heuristic-R. They are prepared by average rows in tables which are used to examine the effects of *n* and *m* for each $T_W$ and $T_U$ combinations. Table 5.12 indicates that $T_W$ and $T_U$ times have no significant effect on the CPU time of first three phases. Moreover, the range of processing times of the jobs also has no effect on the CPU time of these phases. It may also be observed that increasing $T_W$ and $T_U$ times and the range of processing times increase the CPU time of Phase 4.

**Table 5.11:** Effects of Phases in the Algorithm Heuristic-R for the Resumable Jobs Case When $T_W$ is 50 and $T_U$ is 10

| | | Phase 1 | Phase 2 | | Phase 3 | | Phase 4 | |
|---|---|---|---|---|---|---|---|---|
| $m$ | $n$ | Average CPU time | Average CPU time | Average Percent Improvement | Average CPU time | Average Percent Improvement | Average CPU time | Average Percent Improvement |
| 2 | 10 | 3.21 | 3.21 | 1.49 | 6.41 | 0.00 | 3.21 | 0.00 |
| | 20 | 3.25 | 3.25 | 1.97 | 6.50 | 0.03 | 3.25 | 0.15 |
| | 50 | 3.13 | 3.13 | 1.25 | 6.25 | 0.26 | 3.14 | 0.07 |
| | 100 | 3.18 | 3.19 | 0.44 | 6.37 | 0.10 | 3.33 | 0.14 |
| | 200 | 3.27 | 3.28 | 0.21 | 6.55 | 0.05 | 4.83 | 0.07 |
| 3 | 10 | 3.21 | 3.21 | 9.32 | 6.42 | 0.00 | 3.21 | 1.40 |
| | 20 | 3.26 | 3.26 | 2.98 | 6.51 | 0.42 | 3.26 | 0.61 |
| | 50 | 3.16 | 3.16 | 3.17 | 6.31 | 0.62 | 3.19 | 0.19 |
| | 100 | 3.17 | 3.18 | 1.48 | 6.36 | 0.24 | 3.62 | 0.21 |
| | 200 | 3.25 | 3.26 | 1.16 | 6.53 | 0.09 | 6.32 | 0.11 |
| 5 | 20 | 3.14 | 3.14 | 12.92 | 6.27 | 0.61 | 3.15 | 3.63 |
| | 50 | 3.20 | 3.20 | 9.39 | 6.41 | 0.59 | 3.34 | 1.24 |
| | 100 | 3.29 | 3.29 | 3.24 | 6.59 | 1.05 | 4.20 | 0.32 |
| | 200 | 3.27 | 3.28 | 2.03 | 6.58 | 0.14 | 11.48 | 0.23 |
| 10 | 50 | 3.09 | 3.09 | 19.33 | 6.18 | 0.88 | 3.35 | 2.34 |
| | 100 | 3.21 | 3.22 | 10.96 | 6.45 | 1.42 | 5.58 | 1.45 |
| | 200 | 3.40 | 3.41 | 8.00 | 6.89 | 0.34 | 29.98 | 0.64 |
| 20 | 100 | 3.29 | 3.30 | 19.31 | 6.64 | 1.18 | 11.25 | 5.40 |
| | 200 | 3.75 | 3.76 | 14.68 | 7.72 | 0.23 | 59.69 | 2.03 |
| Average | | 3.25 | 3.25 | 6.49 | 6.52 | 0.43 | 8.91 | 1.06 |

Increasing $T_U$ time improves the performance of Phase 2 of the algorithm Heuristic-R. Percent improvement of Phase 2 shows a random pattern when $T_W$ changes. Percent improvement of Phase for problem sets with $p_j \sim U[1, 50]$ is better as compared to the problem sets with $p_j \sim U[1, 10]$.

$T_W$ and $T_U$ times have no effect on the performance of Phase 3. Performance of Phase 4 shows little imrovement when $T_U$ times increase. We also observe that $T_W$ times have no significant effect on the performance of Phase 4. Moreover, Table 5.12 indicates that the performance of Phases 3 and 4 are approximately same for the problem sets with $p_j \sim U[1, 10]$ and $p_j \sim U[1, 50]$.

**Table 5.12:** Performance of Phases in the Algorithm Heuristic-R for the Resumable Jobs Case When the $T_W$ and $T_U$ Changes

| | $T_W$ | $T_U$ | Phase 1 | Phase 2 | | Phase 3 | | Phase 4 | |
|---|---|---|---|---|---|---|---|---|---|
| | | | Average CPU time | Average CPU time | Average Percent Improvement | Average CPU time | Average Percent Improvement | Average CPU time | Average Percent Improvement |
| $U$[1, 10] | 10 | 2 | 2.84 | 2.84 | 4.46 | 2.85 | 0.48 | 4.79 | 0.97 |
| | | 5 | 3.59 | 3.60 | 5.85 | 3.62 | 0.39 | 6.91 | 1.01 |
| | | 10 | 3.52 | 3.53 | 6.15 | 3.54 | 0.47 | 7.50 | 1.41 |
| | 20 | 4 | 3.58 | 3.59 | 4.85 | 3.60 | 0.49 | 5.18 | 0.78 |
| | | 10 | 3.56 | 3.57 | 5.85 | 3.58 | 0.64 | 5.52 | 0.75 |
| | | 20 | 3.58 | 3.59 | 7.10 | 3.60 | 0.65 | 5.74 | 1.06 |
| | 30 | 6 | 2.82 | 2.83 | 5.00 | 2.84 | 0.53 | 4.54 | 0.70 |
| | | 15 | 3.65 | 3.66 | 5.59 | 4.71 | 0.49 | 5.90 | 0.71 |
| | | 30 | 3.81 | 3.81 | 6.59 | 3.83 | 0.37 | 6.03 | 0.62 |
| $U$[1, 50] | 50 | 10 | 3.25 | 3.25 | 6.49 | 6.52 | 0.43 | 7.96 | 1.06 |
| | | 25 | 3.22 | 3.22 | 7.57 | 3.24 | 0.37 | 9.56 | 1.19 |
| | | 50 | 3.25 | 3.25 | 8.72 | 3.28 | 0.40 | 9.85 | 1.30 |
| | 100 | 20 | 3.18 | 3.18 | 5.71 | 3.21 | 0.36 | 9.28 | 1.09 |
| | | 50 | 3.17 | 3.18 | 6.44 | 3.20 | 0.35 | 9.60 | 1.15 |
| | | 100 | 3.15 | 3.15 | 7.01 | 3.18 | 0.37 | 9.86 | 1.27 |
| | 150 | 30 | 3.17 | 3.18 | 6.49 | 3.20 | 0.44 | 10.46 | 1.17 |
| | | 75 | 3.18 | 3.19 | 6.99 | 3.21 | 0.35 | 9.33 | 1.23 |
| | | 150 | 3.16 | 3.16 | 7.74 | 3.19 | 0.29 | 9.34 | 1.20 |
| Average | | | 3.32 | 3.32 | 6.37 | 3.58 | 0.44 | 7.62 | 1.04 |

Figure 5.2 shows the effects of the phases in the algorithm Heuristic-R. We observe that the greatest average improvement is achieved by Phase 2, and the average improvement by Phase 4 is better than the effect of the Phase 3. The average of percent improvement of Phases 2, 3 and 4 in the heuristic algorithm is 6.37, 0.44, and 1.04, respectively.

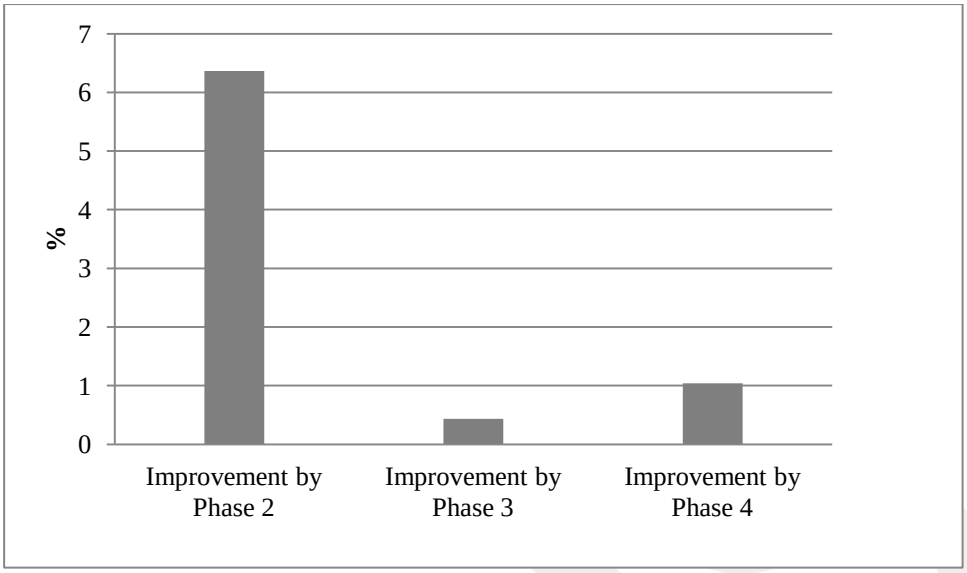


**Figure 5.2:** Percent improvement on the makespan obtained by the phases of the algorithm Heuristic-R for resumable jobs case

To summarize the effects of phases in the algorithm Heuristic-R, we observe that the first three phases require very small CPU times, and the maximum improvement is achieved by Phase 2. Therefore, a user can employ the first three phases to find solution in very short times, or use all phases to get better solutions in little more time.

# CHAPTER 6

# CONCLUSION

In this thesis, we consider a makespan minimization problem of scheduling independent jobs on unrelated parallel machines with machine availability and eligibility constraints. We investigate the problem for two cases: non-resumable and resumable jobs cases.

For both non-resumable and resumable jobs cases, we develop mixed integer linear programming models with lower and upper bounds on the makespan to improve the performance of the MILP models and propose multi-phase heuristic algorithms.

We observe from our experiments that the proposed algorithms developed for both non-resumable and resumable jobs cases find promising results as they solve small- and medium-sized problem instances optimally and find near-optimal solutions for large-sized instances in very short time. The results also reveal that solving the problem by a standard MILP solver seems not to be a useful alternative, especially for solving large-sized problem instances. Thus, solving problems by the heuristic algorithms is a useful alternative, especially for solving large-sized problems.

We also observe that increasing the maximum working time for the non-resumable jobs case has significant positive effect on the performance of the solution approaches MILP and the proposed heuristic algorithm. Furthermore, we observe that increasing

the range of processing times makes difficult to solve the problem for both non-resumable and resumable jobs cases.

In order to guide user, we also discuss the effects of the phases in the proposed heuristic algorithms. We observe from our experiments that the computational times of the first three phases are very small, and solution quality is mostly improved in Phase 2. Although the effect of Phase 4 is quite satisfactory, it requires little more computational time. Therefore, one can use the first three phases to get good results in very short time or can use all phases to get better results in expense of little more computational effort.

There are several extensions of this study which are open for future investigation. An extension would be the study of the same problems considered in this paper for other measures of performance such as mean flow time or maximum lateness. Another future research issue may be the study of same problems with a constraint on the overlapping of unavailability periods on different machines when there is only one worker (or a group of workers) to do the maintenance tasks on all machines. In addition, future research can be directed toward the investigation of the problem under consideration for semi-resumable jobs case.

# REFERENCES


1. **Akturk, M.S., Chosh, J.B. and Gunes E.D.** (2004), Scheduling with tool changes to minimize total completion time: Basic results and SPT performance, *European Journal of Operational Research*, 784–790. Vol. 157.

2. **Alagoz, O. and Azizoglu, M.** (2003), Rescheduling of identical parallel machines under machine eligibility constraints, *European Journal of Operational Research*, 523–532, Vol. 149.

3. **Centeno, G. and Armacost, R.L.** (1997), Parallel machine scheduling with release time and machine eligibility restrictions, *Computers and Industrial Engineering*, 273-276, Vol. 33.

4. **Berrichi, A., Amodeo, L., Yalaoui, F., Chatelet, E. and Mezghiche, M.** (2008), Bi-objective optimization algorithms for joint production and maintenance scheduling: application to the parallel machine problem, *Journal of Intelligent Manufacturing*, 389-400, Vol. 20.

5. **Blazewicz, J., Drozdowski, M., Formanowicz, P., Kubiak, W., and Schmidt, G.** (2000), Scheduling preemptable tasks on parallel processors with limited availability, *Parallel Computing*, 1195–1211, Vol. 26.

6. **Chang, S.Y., and Hwang, H.C.** (1999), The-worst case analysis of the MULTIFIT algorithm for scheduling non-simultaneous parallel machines, *Discrete Applied Mathematics*, 135–147, Vol. 92.

7. **Diedrich, F. and Jansen, K.** (2009), Improved approximation algorithms for scheduling with fixed jobs, Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, 675–684.

8. **Edis, E. B. and Ozkarahan, I.** (2011), A combined integer/constraint programming approach to a resource-constrained parallel machine scheduling

problem with machine eligibility restrictions, *Engineering Optimization*, 135-157, Vol. 43.

9. **Fleszar, K. and Charalambous, C.** (2011), Average-weight-controlled bin-oriented heuristics for the one-dimensional bin-packing problem, *European Journal of Operational Research*, 176–184, Vol. 210.

10. **Garey, M.R. and Johnson, D.S.** (1979), *Computers and intractability: a guide to the theory of NP-completeness*. San Francisco. W.H. Freeman.

11. **Gharbi, A. and Haouari, M.** (2005), Optimal parallel machines scheduling with availability constraints, *Discrete Applied Mathematics*, 63-87, Vol. 148.

12. **Glass, C.A. and Kellerer, H.** (2007), Parallel machine scheduling with job assignment restrictions, *Naval Research Logistics*, 250–257, Vol. 54.

13. **Grigoriu, L. and Friesen, D.K.** (2010), Scheduling on same-speed processors with at most one downtime on each machine, *Discrete Optimization*, 212-221, Vol. 7.

14. **Gupta J.N.D., Ho, J.C.,** (1999), A new heuristic for the one-dimensional bin-packing problem, *Production Planning and Control*, 598-603, Vol. 10.

15. **Hwang, H.C., Lee, K. and Chang. S.Y** (2005), The effects of machine availability on the worst-case performance of LPT, *Discrete Applied Mathematics*, 49–61, Vol. 148.

16. **Hwang, H.C. and Chang. S.Y** (1998), Parallel machines scheduling with machine shutdowns, *Computers and Mathematics with Applications*, 21–31, Vol. 36.

17. **Lee, C.Y.** (1991), Parallel machine scheduling with non-simultaneous machine available time, *Discrete Applied Mathematics*, 53–61, Vol. 30.

18. **Lee, C.Y.** (1996), Machine scheduling with an availability constraint, *Journal of Global Optimization*, 363–382, Vol. 9.

19. **Lee, C.Y., Lei, L. and Pinedo, ML.** (1997), Current trends in deterministic scheduling. *Annals Operations Research*, 1–41,Vol. 70.

20. **Lee, W.C. and Wu, C.C.** (2008), Multi-machine scheduling with deteriorating jobs and scheduled maintenance, *Applied Mathematical Modelling*, 362–373, Vol. 32.

21. **Lenstra, JK, Shmoys, DB. and Tardos, E.** (1990), Approximation algorithms for scheduling unrelated parallel machines, *Mathematical Programming*, 259–271, Vol. 46.

22. **Leung, J.Y.T and Li, C.L.** (2008), Scheduling with processing set restrictions: A survey, *International Journal of Production Economics*, 251-262, Vol. 116.

23. **Li, CL.** (2006), Scheduling unit-length jobs with machine eligibility restrictions, *European Journal of Operational Research,* 1325–1328, Vol. 174.

24. **Liao, CJ., Shyur, DL. and Lin, CH.** (2005), Makespan minimization for two parallel machines with an availability constraint, *European Journal of Operational Research,* 445–456, Vol. 160.

25. **Liao, L. W. and Sheen, G. J.** (2008), Parallel machine scheduling with machine availability and eligibility constraints, *European Journal of Operational Research,* 458–467, Vol. 184.

26. **Lin, CH. and Liao, CJ.** (2007), Makespan minimization for two parallel machines with an unavailable period on each machine, *International Journal of Advanced Manufacturing Technology* 1024–1030, Vol. 33.

27. **Liu, Z., and Sanlaville, E.** (1995), Preemptive scheduling with variable profile, precedence constraints and due dates, *Discrete Applied Mathematics*, 253–280, Vol. 58.

28. **Sanlaville, E. and Schmidt, G.** (1998), Machine scheduling with availability constraints, *Acta Informatica*, 795–811, Vol. 35.

29. **Shchepin, E.V. and Vakhania, N.,** (2005), An optimal rounding gives a better approximation for scheduling unrelated machines, *Operations Research Letters*,127–133, Vol. 33.

30. **Sun, K. and Li, H.** (2010), Scheduling problems with multiple maintenance activities and non-preemptive jobs on two identical parallel machines, *International Journal of Production Economics*, 151–158, Vol. 124.

31. **Suresh, V. and Chaudhuri, D.** (1996), Scheduling of unrelated parallel machines when machine availability is specified, *Production Planning and Control*, 393–400, Vol. 7.

32. **Vairaktarakis, G.L. and Cai, X.,** (2003), The value of processing flexibility in multipurpose machines, *IIE Transactions*, 763–774, Vol. 35.

33. **Xu, D., Cheng, Z., Yin, Y. and Li, H.** (2009), Makespan minimization for two parallel machines scheduling with a periodic availability constraint, *Computers and Operations Research*, 1809–1812, Vol. 36.

34. **Xu, D., Sun, K. and Li, H.** (2008), Parallel machine scheduling with almost periodic maintenance and non-preemptive jobs to minimize makespan, *Computers and Operations Research*, 1344–1349, Vol. 35.

35. **Xu, D., Yin, Y. and Li, H.** (2010), Scheduling jobs under increasing linear machine maintenance time, *Journal of Scheduling*, 443–449, Vol. 13.

36. **Ying, M., Chu, C. and Zuo, C.** (2009), A survey of scheduling with deterministic machine availability constraints, *Computers and Industrial Engineering*, 199–211, Vol. 58.

37. **Yong, H.** (2000), Uniform machine scheduling with machine available constraints, Acta Mathematicae Applicatae Sinica, 122-129, Vol. 16.

# APPENDIX A

## SAWMBS Algorithm by Fleszar and Charalambous (2011)

In this appendix, we introduce the SAWMBS algorithm which is proposed by Fleszar and Charalambous (2011) for one-dimensional bin-packing problem. Maximum continuous working time $T_W$ in our problem can be considered as bin capacity $c$ and the processing time of a job can be considered as the weight of an item for the bin-packing problem. They propose a mechanism of controlling the average weight of items packed in bins. The mechanism is called *sufficient average weight* (SAW) principle which is a procedure of selecting the bin having sufficient average weight. In bin-packing problems, large items cannot be efficiently combined in bins. Therefore, if large items are used in the initial bins, then small items can be used in the later bins. SAWMBS algorithm is the combination of SAW principle and the minimum-bin-slack (MBS) heuristic algorithm which is introduced by Gupta and Ho (1999). For each bin, the MBS heuristic considers all maximal subsets of unpacked items and packs the subset that leaves the smallest slack, which is the difference between the bin capacity and the sum of the weights for all items assigned to that bin.

Flezsar and Charalambous (2011) explain the efficiency of SAW principle with a numerical example. A bin-packing problem is considered where the bin capacity is 15 and there are 8 items with weights (7, 6, 6, 6, 5, 5, 4, 4). First-fit-decreasing (FFD) algorithm pack the items in four bins with weights (7, 6), (6, 6), (5, 5, 4), and (4). MBS also packs the items in a different way where four bins with weights (7, 4, 4), (6, 6), (6, 5), and (5) are enough. The SAW principle for MBS heuristic starts with considering the average weight of bins. The average weight of all items is 5.375 (i.e., (7 + 6 + 6 + 6 + 5 + 5 + 4 + 4)/8 = 5.375). Table A-1 represents the iteration results of the SAWMBS algorithm. Initially, there are three subsets with $\vec{w}(A_1) \in \{7, 6\}$, $\vec{w}(A_2) \in \{7, 5\}$, $\vec{w}(A_3) \in \{7, 4, 4\}$, which are feasible and maximal to be assigned to first bin. A subset of items is feasible if its total weight does not exceed the bin capacity, and maximal if it is feasible but adding any of the currently unpacked items would make it infeasible. The MBS algorithm first selects the subset $A_3$ among all

three subsets for assigning it to the first bin since the subset of $A_3$ has the minimum slack. The item weights in the first bin are 7, 4, and 4, and their average is 5 (i.e., (7+4+4)/3=5). SAW principle blocks the assignment of these items to the first bin since their average is less than the average of all items. Then the SAWMBS considers the subset $A_1$ which has the second minimum slack and an average of 6.5. These items are assigned to the first bin since the average is greater than the average of all items. Then, SAWMBS packs the second bin with items having weights (6, 5, 4) among the remaining items, and the average weight of these items is 5 which is equal to the average of all remaining items (i.e., (6 + 6 + 5 + 5 + 4 + 4)/6 = 5). Therefore, these items are assigned to the second bin, and remaining items can be considered for the third bin. As a result, the SAWMBS packs three bins with item weights (7, 6), (6, 5, 4), and (6, 5, 4).

**Table A-1:** Iteration Results of the SAWMBS Algorithm

| Bin Number | Iteration Number | Items | Item Weights | Slack | Is It Maximal | Average Weight | Items Assigned by SAWMBS | Item Weights Assigned by SAWMBS |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 7 | 8 | No | | 1, 2 | 7, 6 |
| | 2 | 1, 2 | 7, 6 | 2 | Yes | 6.5 | | |
| | 3 | 1, 5 | 7, 5 | 3 | Yes | 6 | | |
| | 4 | 1, 7 | 7, 4 | 4 | No | | | |
| | 5 | 1, 7, 8 | 7, 4, 4 | 0 | Yes | 5 | | |
| 2 | 1 | 3 | 6 | 9 | No | | 3, 5, 7 | 6, 5, 4 |
| | 2 | 3, 4 | 6, 6 | 3 | Yes | 6 | | |
| | 3 | 3, 5 | 6, 5 | 4 | No | | | |
| | 4 | 3, 5, 7 | 6, 5, 4 | 0 | Yes | 5 | | |
| 3 | 1 | 4 | 6 | 9 | No | | 4, 6, 8 | 6, 5, 4 |
| | 2 | 4, 6 | 6, 5 | 4 | No | | | |
| | 3 | 4, 6, 8 | 6, 5, 4 | 0 | Yes | 5 | | |

When we consider these items and their weights as jobs and processing times, respectively the application of the bin-packing heuristics FFD, MBS and SAWMBS give the solutions illustrated by the schedules in Figure A.1.

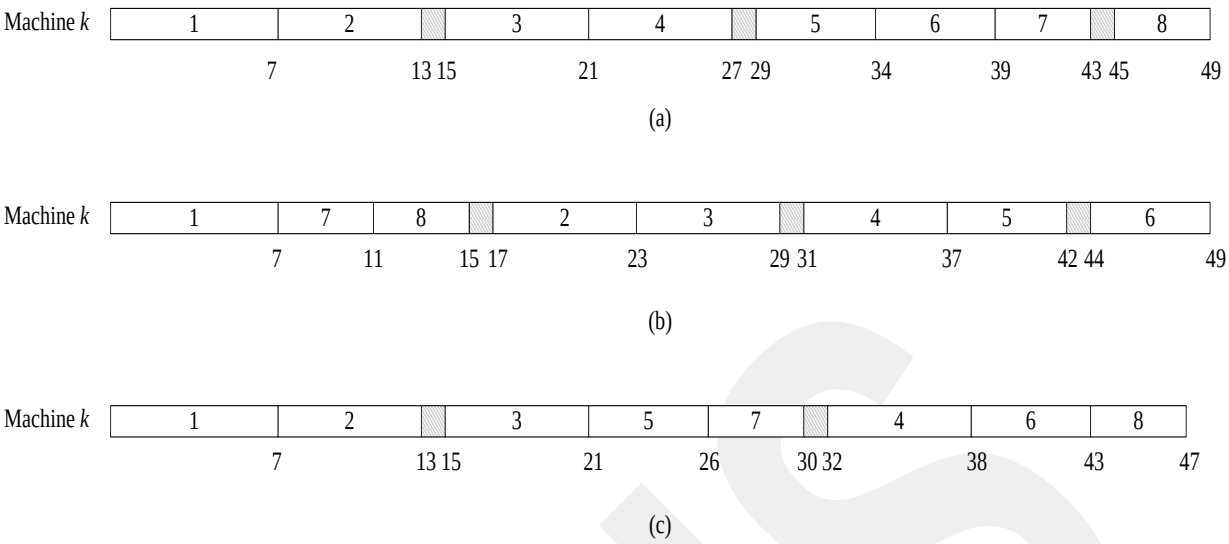


**Figure A-1:** Schedules Obtained by the Bin-Packing Heuristics: (a) FFD, (b) MBS, and (c) SAWMBS

Fleszar and Charalambous (2011) also test the performance of the bin-packing heuristic algorithms on a set of 1587 benchmark problem instances. Among these problems, FFD algorithm with reduction methods to simplify the instances finds the best result for 792 instances in average of 0.08 milliseconds, MBS algorithm with reduction methods finds the best result for 1104 instances in average of 0.67 milliseconds, and SAWMBS algorithm with reduction methods finds the best result for 1385 instances in average of 0.12 milliseconds. The results show that, SAWMBS algorithm outperforms all heuristics, in terms of average solution quality.

Table A-2 provides the notation used in the pseudo-code of the recursive subset-sum (SS) procedure of the SAWMBS heuristic proposed by Flezsar and Charalambous (2011).

**Table A-2:** Notation Used in the Paper by Flezsar and Charalambous (2011)

| Symbol | Description |
|---|---|
| $c$ | bin capacity (maximum working time $T_W$) |
| $N$ | set of all items, $N$= {1,…,$n$} |
| $w_i$ | weight of item $i$ (process time of job) |
| $s_i$ | slack in the bin in which item $i$ is packed |
| $L$ | list of currently unpacked items renumbered such that their weights are non-increasing, $L$ = (1,…, \|$L$\|) (list of assigned jobs) |
| $A$ | feasible subset of currently selected items from $L$ |
| $A$* | incumbent subset |
| $w(A)$ | total weight of items in $A$, $w(A) = \sum_{i \in A} w_i$ |
| $\overline{w}(A)$ | average weight of items in $A$, $\overline{w}(A) = \frac{w(A)}{\|A\|}$ |
| $j$ | item currently considered for addition to $A$, $j \in L$ |
| $p_{min}$ | minimum number of items that must be added to $A$ for it to be maximal |
| $p_{max}$ | maximum number of items that can be added to $A$ for it to improve the incumbent |
| $\overline{w}_{\max}$ | maximum average weight if at least $p_{min}$ items are added to $A$ |
| $s_{min}$ | minimum slack if at most $p_{max}$ items are added to $A$ |

The pseudo-code of the recursive subset-sum (SS) procedure of the SAWMBS heuristic finds a subset for a bin. Before the application of the SS procedure, all unpacked items are renumbered with non-increasing order and listed in $L$. After the application of the SS procedure, the assigned items are removed from the list $L$ and the SS procedure is repeated until $L$ becomes empty.

**Algorithm** Procedure SAWMBSSS

**Procedure** SAWMBSSS($j$)

**if** $j > |L|$ **or** $w_{|L|} > \mathrm{s}(A)$ **then**

/* No items can be added to $A$, update the incumbent and backtrack */

**if** $A$ is not maximal **then return**;

**if** $\overline{w}(A^*) \geq \overline{w}(L)$ then /* if incumbent has sufficient average weight */

**if** $\overline{w}(A) \geq \overline{w}(L)$ **and** $s(A) < s(A^*)$ **then** $A^* \leftarrow A$;

**end**

**else if** $\overline{w}(A) \geq \overline{w}(A^*)$ **or** $(\overline{w}(A) = \overline{w}(A^*)$ **and** $s(A) < s(A^*))$ **then** $A^* \leftarrow A$;

**return;**

**while** $w_j > \mathrm{s}(A)$ **do** $j \leftarrow j+1$ /* skip too large items */

**while** $j \leq |L|$ **do** /* reductions */

**if** $\overline{w}(A^*) \geq \overline{w}(L)$ **and** $\mathrm{s}(\mathrm{A}) - \mathrm{w}(\{j,\ldots, |L|\}) \geq s(A^*)$ **then return**;

**if** subset $A \cup \{j,\ldots, |L|\}$ would be feasible but not maximal **then return;**

**/*** additional SAW-related reductions */

$$p_{\min} \leftarrow \min\left\{\left\lfloor \frac{s(A)}{w_j} \right\rfloor, |L| - j + 1\right\};$$

$$\overline{w}_{\max} \leftarrow \frac{w(A) + p_{\min} w_j}{|A| + p_{\min}};$$

$$p_{\max} \leftarrow \left\lfloor \frac{c}{\min\{\overline{w}(L), \overline{w}(A^*)\}} \right\rfloor - |A|;$$

$$s_{\min} \leftarrow \max\{s(A) - p_{\max} w_j, 0\};$$

**if** $p_{\min} > p_{\max}$ **then return;**

**if** $\overline{w}(A^*) \geq \overline{w}(L)$ then /* if incumbent has sufficient average weight */

**if** $\overline{w}_{\max} < \overline{w}(L)$ **or** ($\overline{w}_{\max} \geq \overline{w}(L)$ **and** $s_{\min} \geq$ s(A*)) **then return;**

**end**

**else if** $\overline{w}_{\max} < \overline{w}(A^*)$ **or** ($\overline{w}_{\max} = \overline{w}(A^*)$ **and** $s_{\min} \geq s(A^*)$) **then return**;

/* Add $j$ to $A$, invoke a recursive call, and then remove $j$ from $A$ */

$A \leftarrow A \cup \{j\}$;

SAWMBSSS( $j$+1);

$A \leftarrow A \setminus \{j\}$;

**if** $\overline{w}(A^*) \geq \overline{w}(L)$ **and** $s(A^*) = 0$ **then return**; /*backtrack if incumbent optimal */

$j \leftarrow j + 1$;

**while** $j \leq |L|$ **and** $w_j = w_{j-1}$ **do** $j \leftarrow j + 1$ /* skip items of same weight */

**end**

**end procedure**

## APPENDIX B Performance of Solution Approaches for Non-Resumable Jobs

**Table B-1:** Performance of the Solution Approaches for the Non-Resumable Jobs Case When $T_W$ is 10 and $T_U$ is 5.

| | | MILP | | | | | | | Heuristic-NR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 1.54 | 1.80 | 1.54 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.61 | 1.80 | 1.61 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 4607.33 | 10801.77 | 297.59 | 6 | 4 | 0.66 | 2.87 | 0.04 | 0.09 | 5 | 1.04 | 3.75 |
| | 100 | 5709.81 | 10801.83 | 8.02 | 5 | 5 | 0.58 | 1.47 | 0.52 | 0.89 | 4 | 0.61 | 1.68 |
| | 200 | 7522.96 | 10802.26 | 412.15 | 3 | 6 | 0.48 | 2.20 | 3.69 | 18.72 | 4 | 0.19 | 0.70 |
| 3 | 10 | 1.76 | 2.03 | 1.76 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.75 | 1.92 | 1.75 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 9 | 1.25 | 12.50 |
| | 50 | 1264.20 | 10801.86 | 151.63 | 9 | 1 | 0.21 | 2.11 | 0.07 | 0.14 | 8 | 0.31 | 2.11 |
| | 100 | 6492.35 | 10802.19 | 19.59 | 4 | 6 | 1.25 | 3.16 | 0.41 | 0.95 | 5 | 1.04 | 2.63 |
| | 200 | 9727.50 | 10802.34 | 62.57 | 1 | 9 | 0.76 | 2.79 | 4.04 | 10.63 | 2 | 0.35 | 1.40 |
| 5 | 20 | 1.93 | 3.34 | 1.93 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 1152.14 | 10801.73 | 17.14 | 9 | 1 | 0.26 | 2.56 | 0.12 | 0.16 | 5 | 1.15 | 2.86 |
| | 100 | 5769.05 | 10801.76 | 90.08 | 5 | 5 | 1.88 | 5.95 | 0.67 | 1.17 | 3 | 1.97 | 7.06 |
| | 200 | 6988.52 | 10802.25 | 270.54 | 4 | 6 | 1.26 | 4.65 | 5.41 | 13.81 | 2 | 0.58 | 1.20 |
| 10 | 50 | 634.41 | 3132.47 | 77.32 | 10 | 0 | 0.00 | 0.00 | 0.16 | 0.22 | 9 | 6.00 | 60.00 |
| | 100 | 2171.42 | 10801.73 | 978.00 | 9 | 1 | 0.42 | 4.17 | 0.90 | 1.33 | 6 | 1.61 | 4.35 |
| | 200 | 3605.35 | 10801.95 | 1180.71 | 8 | 2 | 1.22 | 8.93 | 7.18 | 11.64 | 4 | 2.90 | 10.91 |
| 20 | 100 | 6.52 | 29.77 | 4.30 | 10 | 0 | 0.00 | 0.00 | 1.27 | 1.98 | 8 | 2.86 | 14.29 |
| | 200 | 3.18 | 4.05 | 3.18 | 10 | 0 | 0.00 | 0.00 | 9.19 | 19.38 | 7 | 1.67 | 5.56 |
| Average | | 2929.65 | 6420.99 | 188.49 | 143(sum) | 46(sum) | 0.47 | 2.15 | 1.77 | 4.27 | 121(sum) | 1.24 | 6.89 |

**Table B-2**: Performance of the Solution Approaches for the Non-Resumable Jobs Case When $T_W$ is 10 and $T_U$ is 10

| | | MILP | | | | | | | Heuristic-NR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimums solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 1.22 | 1.78 | 1.22 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.02 | 10 | 0.00 | 0.00 |
| | 20 | 1.28 | 1.70 | 1.28 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 50 | 1157.85 | 10801.19 | 77.39 | 9 | 1 | 0.57 | 5.68 | 0.05 | 0.13 | 8 | 0.96 | 5.24 |
| | 100 | 5454.35 | 10801.48 | 104.68 | 5 | 5 | 0.80 | 2.77 | 0.80 | 1.55 | 5 | 0.75 | 2.22 |
| | 200 | 9727.41 | 10802.09 | 875.92 | 1 | 9 | 0.76 | 3.21 | 8.65 | 57.75 | 5 | 0.28 | 1.10 |
| 3 | 10 | 1.59 | 1.75 | 1.59 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.59 | 1.70 | 1.59 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 9 | 2.00 | 20.00 |
| | 50 | 1380.27 | 10801.75 | 279.89 | 9 | 1 | 0.16 | 1.60 | 0.11 | 0.20 | 8 | 0.18 | 1.60 |
| | 100 | 5419.14 | 10801.84 | 33.70 | 5 | 5 | 1.21 | 4.40 | 0.57 | 1.39 | 5 | 1.17 | 4.00 |
| | 200 | 8439.54 | 10802.17 | 524.94 | 2 | 7 | 0.41 | 1.51 | 7.23 | 24.78 | 2 | 0.39 | 1.72 |
| 5 | 20 | 1.89 | 2.58 | 1.89 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 1286.37 | 10801.77 | 58.75 | 9 | 1 | 0.46 | 4.62 | 0.13 | 0.23 | 6 | 1.00 | 4.62 |
| | 100 | 4887.73 | 10801.86 | 1063.98 | 6 | 4 | 1.63 | 12.04 | 0.86 | 2.02 | 5 | 1.79 | 10.00 |
| | 200 | 8110.54 | 10802.09 | 1440.46 | 4 | 6 | 0.39 | 0.90 | 8.05 | 24.44 | 2 | 0.48 | 0.90 |
| 10 | 50 | 423.98 | 2150.34 | 57.02 | 10 | 0 | 0.00 | 0.00 | 0.18 | 0.25 | 9 | 11.00 | 110.00 |
| | 100 | 69.98 | 485.02 | 22.08 | 10 | 0 | 0.00 | 0.00 | 1.02 | 1.67 | 7 | 0.96 | 3.57 |
| | 200 | 3658.57 | 10801.44 | 557.09 | 7 | 3 | 3.10 | 15.71 | 7.70 | 11.69 | 4 | 3.68 | 15.71 |
| 20 | 100 | 12.82 | 43.39 | 7.43 | 10 | 0 | 0.00 | 0.00 | 1.30 | 1.94 | 8 | 2.86 | 14.29 |
| | 200 | 2.78 | 4.09 | 2.78 | 10 | 0 | 0.00 | 0.00 | 9.18 | 16.54 | 7 | 1.30 | 4.35 |
| Average | | 2633.63 | 5826.84 | 269.14 | 147(sum) | 42(sum) | 0.50 | 2.76 | 2.41 | 7.61 | 130(sum) | 1.52 | 10.49 |

**Table B-3**: Performance of the Solution Approaches for the Non-Resumable Jobs Case When $T_W$ is 20 and $T_U$ is 4

| | | MILP | | | | | | | Heuristic-NR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 1.52 | 1.67 | 1.52 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.02 | 10 | 0.00 | 0.00 |
| | 20 | 1.67 | 1.75 | 1.67 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 1.58 | 1.73 | 1.58 | 10 | 0 | 0.00 | 0.00 | 0.02 | 0.05 | 9 | 0.08 | 0.77 |
| | 100 | 6.37 | 31.16 | 3.47 | 10 | 0 | 0.00 | 0.00 | 0.29 | 0.64 | 7 | 0.11 | 0.36 |
| | 200 | 4325.82 | 10801.97 | 12.91 | 6 | 4 | 0.19 | 0.88 | 1.32 | 3.17 | 7 | 0.12 | 0.70 |
| 3 | 10 | 1.52 | 1.80 | 1.52 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.63 | 1.81 | 1.63 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 1.77 | 2.11 | 1.77 | 10 | 0 | 0.00 | 0.00 | 0.04 | 0.08 | 9 | 0.13 | 1.27 |
| | 100 | 1083.47 | 10801.64 | 3.25 | 9 | 1 | 0.17 | 1.74 | 0.32 | 0.55 | 6 | 0.24 | 0.64 |
| | 200 | 4390.52 | 10801.92 | 14.58 | 6 | 4 | 0.17 | 0.69 | 2.95 | 6.58 | 6 | 0.17 | 0.69 |
| 5 | 20 | 1.54 | 1.72 | 1.54 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 27.30 | 148.17 | 27.30 | 10 | 0 | 0.00 | 0.00 | 0.09 | 0.09 | 6 | 1.13 | 3.45 |
| | 100 | 2285.54 | 10801.78 | 124.10 | 8 | 2 | 0.30 | 1.49 | 0.55 | 0.86 | 5 | 1.33 | 7.35 |
| | 200 | 2178.01 | 10801.84 | 10.72 | 8 | 2 | 0.14 | 0.69 | 4.78 | 14.05 | 3 | 0.51 | 0.76 |
| 10 | 50 | 1.83 | 3.86 | 1.83 | 10 | 0 | 0.00 | 0.00 | 0.10 | 0.13 | 9 | 1.00 | 10.00 |
| | 100 | 146.25 | 1031.17 | 13.58 | 10 | 0 | 0.00 | 0.00 | 0.80 | 1.17 | 7 | 1.51 | 5.56 |
| | 200 | 1317.88 | 10801.70 | 25.45 | 9 | 1 | 0.20 | 2.04 | 6.31 | 9.56 | 5 | 2.30 | 11.36 |
| 20 | 100 | 1.71 | 1.83 | 1.71 | 10 | 0 | 0.00 | 0.00 | 1.38 | 2.08 | 7 | 4.29 | 14.29 |
| | 200 | 1.85 | 1.97 | 1.85 | 10 | 0 | 0.00 | 0.00 | 8.69 | 15.42 | 6 | 3.08 | 7.69 |
| Average | | 830.41 | 3475.87 | 13.26 | 176(sum) | 14(sum) | 0.06 | 0.40 | 1.46 | 2.87 | 142(sum) | 0.84 | 3.41 |

**Table B-4**: Performance of the Solution Approaches for the Non-Resumable Jobs Case When $T_W$ is 20 and $T_U$ is 10

| | | MILP | | | | | | | Heuristic-NR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 1.47 | 1.63 | 1.47 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.02 | 10 | 0.00 | 0.00 |
| | 20 | 1.66 | 1.77 | 1.66 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 50 | 1.84 | 2.59 | 1.84 | 10 | 0 | 0.00 | 0.00 | 0.02 | 0.05 | 9 | 0.06 | 0.63 |
| | 100 | 8.69 | 33.66 | 2.74 | 10 | 0 | 0.00 | 0.00 | 0.35 | 1.02 | 7 | 0.09 | 0.29 |
| | 200 | 3336.97 | 10801.98 | 31.74 | 7 | 3 | 0.20 | 1.26 | 1.74 | 5.92 | 8 | 0.16 | 1.40 |
| 3 | 10 | 1.61 | 1.91 | 1.61 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.65 | 1.86 | 1.65 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 2.81 | 11.73 | 2.81 | 10 | 0 | 0.00 | 0.00 | 0.05 | 0.16 | 9 | 0.10 | 1.03 |
| | 100 | 2164.03 | 10801.92 | 4.33 | 8 | 2 | 0.56 | 5.08 | 0.35 | 0.66 | 6 | 0.20 | 0.52 |
| | 200 | 3299.32 | 10802.00 | 24.35 | 7 | 3 | 0.08 | 0.26 | 6.44 | 21.77 | 6 | 0.10 | 0.28 |
| 5 | 20 | 1.57 | 1.77 | 1.57 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 1.98 | 2.85 | 1.98 | 10 | 0 | 0.00 | 0.00 | 0.09 | 0.11 | 6 | 0.96 | 2.86 |
| | 100 | 4350.81 | 10801.92 | 3.29 | 6 | 4 | 1.73 | 13.75 | 86.58 | 859.95 | 5 | 1.87 | 13.75 |
| | 200 | 2178.70 | 10801.80 | 16.68 | 8 | 2 | 0.11 | 0.55 | 7.57 | 22.72 | 3 | 0.41 | 0.62 |
| 10 | 50 | 1.78 | 3.22 | 1.78 | 10 | 0 | 0.00 | 0.00 | 0.10 | 0.14 | 9 | 1.00 | 10.00 |
| | 100 | 268.15 | 1222.06 | 143.31 | 10 | 0 | 0.00 | 0.00 | 0.92 | 1.38 | 7 | 1.43 | 5.56 |
| | 200 | 50.08 | 236.70 | 26.37 | 10 | 0 | 0.00 | 0.00 | 7.22 | 11.11 | 5 | 3.01 | 22.00 |
| 20 | 100 | 1.75 | 1.91 | 1.75 | 10 | 0 | 0.00 | 0.00 | 1.38 | 2.08 | 7 | 4.29 | 14.29 |
| | 200 | 1.86 | 2.00 | 1.86 | 10 | 0 | 0.00 | 0.00 | 8.59 | 15.39 | 6 | 3.08 | 7.69 |
| Average | | 825.09 | 2922.91 | 14.36 | 176(sum) | 14(sum) | 0.14 | 1.10 | 6.39 | 49.60 | 143(sum) | 0.88 | 4.26 |

**Table B-5:** Performance of the Solution Approaches for the Non-Resumable Jobs Case When $T_W$ is 20 and $T_U$ is 20

| | | MILP | | | | | | | Heuristic-NR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 1.50 | 1.61 | 1.50 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.02 | 10 | 0.00 | 0.00 |
| | 20 | 1.56 | 1.70 | 1.56 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 1.80 | 2.22 | 1.80 | 10 | 0 | 0.00 | 0.00 | 0.02 | 0.03 | 9 | 0.05 | 0.48 |
| | 100 | 7.71 | 25.77 | 3.11 | 10 | 0 | 0.00 | 0.00 | 0.35 | 1.02 | 7 | 0.06 | 0.22 |
| | 200 | 3606.22 | 10801.94 | 39.66 | 7 | 3 | 0.21 | 1.91 | 1.74 | 5.92 | 8 | 0.21 | 2.01 |
| 3 | 10 | 1.81 | 2.69 | 1.81 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.60 | 1.78 | 1.60 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 23.17 | 215.41 | 22.99 | 10 | 0 | 0.00 | 0.00 | 0.05 | 0.14 | 9 | 0.08 | 0.79 |
| | 100 | 1085.12 | 10801.77 | 4.23 | 9 | 1 | 0.08 | 0.80 | 0.35 | 0.67 | 6 | 0.15 | 0.40 |
| | 200 | 3312.70 | 10801.95 | 18.62 | 7 | 3 | 0.10 | 0.40 | 6.47 | 21.78 | 6 | 0.12 | 0.40 |
| 5 | 20 | 1.66 | 1.75 | 1.66 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 17.69 | 114.72 | 17.58 | 10 | 0 | 0.00 | 0.00 | 0.09 | 0.09 | 6 | 0.78 | 2.22 |
| | 100 | 4333.71 | 10801.86 | 203.38 | 6 | 4 | 4.80 | 22.45 | 0.70 | 1.81 | 5 | 2.86 | 21.00 |
| | 200 | 5417.87 | 10801.97 | 18.33 | 5 | 5 | 0.26 | 0.92 | 7.56 | 22.72 | 1 | 0.45 | 0.92 |
| 10 | 50 | 1.73 | 3.95 | 1.73 | 10 | 0 | 0.00 | 0.00 | 0.10 | 0.13 | 9 | 1.00 | 10.00 |
| | 100 | 27.31 | 177.55 | 6.64 | 10 | 0 | 0.00 | 0.00 | 0.95 | 1.55 | 7 | 1.36 | 5.56 |
| | 200 | 107.70 | 803.20 | 14.42 | 10 | 0 | 0.00 | 0.00 | 7.21 | 11.09 | 5 | 4.16 | 35.00 |
| 20 | 100 | 1.87 | 3.08 | 1.87 | 10 | 0 | 0.00 | 0.00 | 1.38 | 2.08 | 7 | 4.29 | 14.29 |
| | 200 | 1.85 | 2.00 | 1.85 | 10 | 0 | 0.00 | 0.00 | 8.75 | 15.38 | 6 | 3.08 | 7.69 |
| Average | | 944.98 | 2914.05 | 19.18 | 174(sum) | 16(sum) | 0.29 | 1.39 | 1.88 | 4.45 | 141(sum) | 0.98 | 5.31 |

**Table B-6:** Performance of the Solution Approaches for the Non-Resumable Jobs Case When $T_W$ is 30 and $T_U$ is 6

| | | MILP | | | | | | | Heuristic-NR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 1.61 | 1.92 | 1.61 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.02 | 10 | 0.00 | 0.00 |
| | 20 | 1.58 | 1.77 | 1.58 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 50 | 1.59 | 1.66 | 1.59 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.03 | 9 | 0.08 | 0.78 |
| | 100 | 1082.19 | 10801.64 | 2.20 | 9 | 1 | 0.04 | 0.36 | 0.36 | 0.81 | 6 | 0.14 | 0.37 |
| | 200 | 2169.00 | 10801.84 | 6.65 | 8 | 2 | 0.11 | 0.87 | 1.38 | 4.14 | 8 | 0.12 | 1.05 |
| 3 | 10 | 1.45 | 1.56 | 1.45 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.55 | 1.70 | 1.55 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 1.64 | 1.94 | 1.64 | 10 | 0 | 0.00 | 0.00 | 0.04 | 0.11 | 9 | 0.13 | 1.27 |
| | 100 | 2.31 | 4.36 | 2.31 | 10 | 0 | 0.00 | 0.00 | 0.28 | 0.50 | 7 | 0.19 | 0.64 |
| | 200 | 2176.27 | 10801.94 | 7.78 | 8 | 2 | 0.13 | 0.66 | 1.78 | 3.45 | 7 | 0.16 | 0.66 |
| 5 | 20 | 1.51 | 1.66 | 1.51 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 9 | 1.54 | 15.38 |
| | 50 | 15.32 | 74.23 | 5.75 | 10 | 0 | 0.00 | 0.00 | 0.10 | 0.13 | 6 | 1.15 | 4.00 |
| | 100 | 49.66 | 310.02 | 46.01 | 10 | 0 | 0.00 | 0.00 | 0.62 | 1.22 | 7 | 1.36 | 10.61 |
| | 200 | 2163.78 | 10801.69 | 4.22 | 8 | 2 | 0.14 | 0.74 | 5.28 | 13.02 | 4 | 0.44 | 0.77 |
| 10 | 50 | 1.55 | 1.73 | 1.55 | 10 | 0 | 0.00 | 0.00 | 0.10 | 0.14 | 9 | 1.00 | 10.00 |
| | 100 | 1.62 | 1.78 | 1.62 | 10 | 0 | 0.00 | 0.00 | 0.59 | 0.75 | 7 | 1.59 | 5.56 |
| | 200 | 10.07 | 51.06 | 10.07 | 10 | 0 | 0.00 | 0.00 | 5.60 | 8.09 | 5 | 1.14 | 2.44 |
| 20 | 100 | 1.75 | 2.14 | 1.75 | 10 | 0 | 0.00 | 0.00 | 1.42 | 2.08 | 7 | 4.29 | 14.29 |
| | 200 | 1.84 | 1.91 | 1.84 | 10 | 0 | 0.00 | 0.00 | 8.48 | 15.44 | 7 | 2.31 | 7.69 |
| Average | | 404.54 | 2298.24 | 5.40 | 183(sum) | 7(sum) | 0.02 | 0.14 | 1.37 | 2.63 | 147(sum) | 0.82 | 3.97 |

**Table B-7:** Performance of the Solution Approaches for the Non-Resumable Jobs Case When $T_W$ is 30 and $T_U$ is 15

| | | MILP | | | | | | | Heuristic-NR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 1.52 | 1.58 | 1.52 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.56 | 1.81 | 1.56 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 1.60 | 1.70 | 1.60 | 10 | 0 | 0.00 | 0.00 | 0.02 | 0.03 | 9 | 0.06 | 0.65 |
| | 100 | 3.20 | 7.66 | 3.20 | 10 | 0 | 0.00 | 0.00 | 0.41 | 1.13 | 7 | 0.09 | 0.30 |
| | 200 | 1128.34 | 10801.81 | 9.05 | 9 | 1 | 0.01 | 0.14 | 1.54 | 5.77 | 8 | 0.03 | 0.16 |
| 3 | 10 | 1.50 | 1.61 | 1.50 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.54 | 1.70 | 1.54 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 1.65 | 1.94 | 1.65 | 10 | 0 | 0.00 | 0.00 | 0.05 | 0.14 | 9 | 0.10 | 1.03 |
| | 100 | 1082.44 | 10801.62 | 2.49 | 9 | 1 | 0.15 | 1.49 | 0.27 | 0.49 | 7 | 0.25 | 1.49 |
| | 200 | 3256.04 | 10801.86 | 5.54 | 7 | 3 | 0.13 | 0.53 | 1.79 | 3.55 | 7 | 0.13 | 0.53 |
| 5 | 20 | 1.55 | 1.77 | 1.55 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 9 | 1.54 | 15.38 |
| | 50 | 3.95 | 8.78 | 3.95 | 10 | 0 | 0.00 | 0.00 | 0.11 | 0.20 | 6 | 1.01 | 4.00 |
| | 100 | 2162.51 | 10801.87 | 2.88 | 8 | 2 | 0.46 | 3.26 | 0.74 | 1.72 | 6 | 2.74 | 21.33 |
| | 200 | 2165.85 | 10801.78 | 6.00 | 8 | 2 | 0.12 | 0.64 | 6.72 | 18.30 | 4 | 0.37 | 0.64 |
| 10 | 50 | 1.56 | 1.73 | 1.56 | 10 | 0 | 0.00 | 0.00 | 0.10 | 0.14 | 9 | 1.00 | 10.00 |
| | 100 | 1.71 | 1.98 | 1.71 | 10 | 0 | 0.00 | 0.00 | 0.59 | 0.74 | 7 | 1.59 | 5.56 |
| | 200 | 4.73 | 8.58 | 4.33 | 10 | 0 | 0.00 | 0.00 | 5.59 | 8.03 | 5 | 0.95 | 2.00 |
| 20 | 100 | 1.74 | 1.88 | 1.74 | 10 | 0 | 0.00 | 0.00 | 1.41 | 2.08 | 7 | 4.29 | 14.29 |
| | 200 | 1.82 | 2.02 | 1.82 | 10 | 0 | 0.00 | 0.00 | 8.48 | 15.44 | 7 | 2.31 | 7.69 |
| Average | | 517.10 | 2844.93 | 2.91 | 181(sum) | 9(sum) | 0.05 | 0.32 | 1.47 | 3.04 | 147(sum) | 0.87 | 4.48 |

**Table B-8:** Performance of the Solution Approaches for the Non-Resumable Jobs Case When $T_W$ is 30 and $T_U$ is 30

| | | MILP | | | | | | | Heuristic-NR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 1.61 | 1.77 | 1.61 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.54 | 1.64 | 1.54 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 50 | 1.63 | 1.97 | 1.63 | 10 | 0 | 0.00 | 0.00 | 0.02 | 0.05 | 9 | 0.05 | 0.50 |
| | 100 | 2.36 | 4.17 | 2.36 | 10 | 0 | 0.00 | 0.00 | 0.41 | 1.14 | 7 | 0.07 | 0.23 |
| | 200 | 2234.33 | 10801.86 | 6.65 | 8 | 2 | 0.02 | 0.12 | 1.54 | 5.77 | 8 | 0.02 | 0.12 |
| 3 | 10 | 1.53 | 1.67 | 1.53 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.60 | 1.73 | 1.60 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 3.80 | 22.86 | 3.80 | 10 | 0 | 0.00 | 0.00 | 0.05 | 0.14 | 9 | 0.08 | 0.79 |
| | 100 | 3.26 | 7.45 | 3.26 | 10 | 0 | 0.00 | 0.00 | 0.28 | 0.50 | 7 | 0.12 | 0.40 |
| | 200 | 4334.61 | 10801.95 | 5.92 | 6 | 4 | 0.12 | 0.40 | 1.80 | 3.55 | 6 | 0.12 | 0.40 |
| 5 | 20 | 1.67 | 1.86 | 1.67 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 9 | 1.54 | 15.38 |
| | 50 | 89.34 | 865.39 | 89.34 | 10 | 0 | 0.00 | 0.00 | 0.11 | 0.20 | 6 | 0.87 | 4.00 |
| | 100 | 3242.41 | 10801.94 | 25.71 | 7 | 3 | 3.45 | 32.58 | 0.73 | 1.72 | 5 | 3.87 | 34.44 |
| | 200 | 1087.38 | 10801.92 | 8.25 | 9 | 1 | 0.10 | 0.96 | 6.72 | 18.31 | 4 | 0.29 | 0.50 |
| 10 | 50 | 1.70 | 1.89 | 1.70 | 10 | 0 | 0.00 | 0.00 | 0.10 | 0.14 | 9 | 1.00 | 10.00 |
| | 100 | 1.78 | 1.95 | 1.78 | 10 | 0 | 0.00 | 0.00 | 0.60 | 0.73 | 7 | 1.06 | 5.56 |
| | 200 | 3.75 | 7.66 | 3.75 | 10 | 0 | 0.00 | 0.00 | 5.59 | 8.03 | 5 | 0.74 | 1.54 |
| 20 | 100 | 1.75 | 2.08 | 1.75 | 10 | 0 | 0.00 | 0.00 | 1.42 | 2.08 | 7 | 4.29 | 14.29 |
| | 200 | 1.85 | 1.95 | 1.85 | 10 | 0 | 0.00 | 0.00 | 8.48 | 15.44 | 7 | 2.31 | 7.69 |
| Average | | 579.89 | 2322.83 | 8.72 | 180(sum) | 10(sum) | 0.19 | 1.79 | 1.47 | 3.04 | 145(sum) | 0.86 | 5.04 |

**Table B-9:** Performance of the Solution Approaches for the Non-Resumable Jobs Case When $T_W$ is 50 and $T_U$ is 25

| | | MILP | | | | | | | Heuristic-NR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 1.63 | 1.81 | 1.63 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.88 | 2.76 | 1.88 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 7743.99 | 10801.73 | 13.83 | 3 | 7 | 1.26 | 5.12 | 0.08 | 0.16 | 3 | 1.24 | 4.96 |
| | 100 | 10801.72 | 10801.83 | 516.11 | 0 | 8 | 1.95 | 3.23 | 1.13 | 2.19 | 0 | 0.97 | 1.98 |
| | 200 | 9010.62 | 10802.33 | 310.64 | 1 | 5 | 1.55 | 3.29 | 11.29 | 20.84 | 0 | 0.44 | 0.97 |
| 3 | 10 | 1.52 | 1.72 | 1.52 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 2.41 | 7.05 | 2.41 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 6539.72 | 10801.98 | 80.85 | 4 | 6 | 1.40 | 4.62 | 0.17 | 0.44 | 2 | 1.56 | 4.62 |
| | 100 | 9453.84 | 10802.06 | 40.34 | 1 | 7 | 2.58 | 6.41 | 1.08 | 2.19 | 0 | 1.02 | 3.09 |
| | 200 | 10801.99 | 10802.16 | 552.04 | 0 | 9 | 1.37 | 4.26 | 7.10 | 11.73 | 0 | 0.52 | 1.77 |
| 5 | 20 | 4.81 | 22.62 | 2.73 | 10 | 0 | 0.00 | 0.00 | 0.02 | 0.03 | 9 | 0.25 | 2.50 |
| | 50 | 819.53 | 4581.92 | 566.37 | 10 | 0 | 0.00 | 0.00 | 0.13 | 0.20 | 5 | 0.39 | 1.63 |
| | 100 | 10801.86 | 10801.92 | 851.30 | 0 | 10 | 2.70 | 9.28 | 1.42 | 3.89 | 0 | 2.86 | 9.28 |
| | 200 | 10801.95 | 10802.09 | 798.04 | 0 | 10 | 3.88 | 6.98 | 7.58 | 9.98 | 0 | 1.56 | 3.86 |
| 10 | 50 | 612.26 | 5968.52 | 48.67 | 10 | 0 | 0.00 | 0.00 | 0.23 | 0.33 | 7 | 0.99 | 5.00 |
| | 100 | 10801.85 | 10802.23 | 504.77 | 0 | 10 | 6.03 | 9.28 | 1.69 | 2.13 | 0 | 5.76 | 10.53 |
| | 200 | 10801.94 | 10802.23 | 1887.00 | 0 | 10 | 9.77 | 16.93 | 16.81 | 22.61 | 0 | 7.72 | 14.91 |
| 20 | 100 | 1086.54 | 10802.02 | 10.27 | 9 | 1 | 1.30 | 13.04 | 4.65 | 6.14 | 4 | 4.35 | 17.39 |
| | 200 | 9808.58 | 10802.15 | 2802.31 | 2 | 8 | 3.24 | 6.67 | 29.39 | 36.99 | 0 | 4.42 | 6.82 |
| Average | | 5784.14 | 7379.53 | 473.30 | 90(sum) | 91(sum) | 1.95 | 4.69 | 4.36 | 6.31 | 70(sum) | 1.79 | 4.70 |

**Table B-10:** Performance of the Solution Approaches for the Non-Resumable Jobs Case When $T_W$ is 50 and $T_U$ is 50

| | | MILP | | | | | | | Heuristic-NR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 1.58 | 2.08 | 1.58 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 2.03 | 3.11 | 2.03 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 8642.47 | 10802.02 | 48.00 | 2 | 8 | 1.40 | 6.12 | 0.10 | 0.30 | 2 | 1.48 | 6.12 |
| | 100 | 10801.78 | 10801.95 | 1402.56 | 0 | 8 | 3.13 | 5.76 | 7.20 | 56.45 | 0 | 1.47 | 2.67 |
| | 200 | 10802.17 | 10802.28 | 977.40 | 0 | 5 | 3.55 | 5.31 | 17.42 | 36.39 | 0 | 0.69 | 1.31 |
| 3 | 10 | 1.18 | 1.33 | 1.18 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.62 | 2.27 | 1.62 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 8651.60 | 10801.61 | 1507.99 | 2 | 8 | 2.88 | 8.57 | 0.25 | 0.88 | 1 | 2.78 | 8.21 |
| | 100 | 10801.40 | 10801.58 | 200.40 | 0 | 10 | 1.91 | 5.29 | 1.74 | 5.11 | 0 | 1.00 | 4.42 |
| | 200 | 10801.75 | 10801.89 | 2154.71 | 0 | 9 | 2.08 | 4.54 | 11.37 | 22.81 | 0 | 0.52 | 2.49 |
| 5 | 20 | 2.80 | 5.77 | 2.80 | 10 | 0 | 0.00 | 0.00 | 0.02 | 0.03 | 9 | 0.19 | 1.90 |
| | 50 | 1389.47 | 6012.76 | 785.41 | 10 | 0 | 0.00 | 0.00 | 0.14 | 0.19 | 5 | 0.31 | 1.28 |
| | 100 | 10801.74 | 10801.83 | 1832.84 | 0 | 10 | 5.12 | 14.90 | 1.52 | 3.97 | 0 | 2.83 | 13.32 |
| | 200 | 10801.99 | 10802.12 | 490.65 | 0 | 10 | 5.46 | 7.73 | 8.42 | 14.84 | 0 | 1.70 | 5.58 |
| 10 | 50 | 978.19 | 9628.23 | 441.55 | 10 | 0 | 0.00 | 0.00 | 0.24 | 0.36 | 6 | 1.16 | 5.00 |
| | 100 | 10801.80 | 10801.97 | 2970.90 | 0 | 10 | 3.92 | 6.67 | 1.72 | 2.13 | 0 | 4.58 | 8.33 |
| | 200 | 10801.92 | 10802.14 | 81.20 | 0 | 10 | 18.59 | 24.69 | 19.09 | 26.13 | 0 | 10.07 | 22.28 |
| 20 | 100 | 1592.50 | 10801.66 | 958.30 | 9 | 1 | 3.00 | 30.00 | 4.64 | 5.80 | 4 | 6.11 | 35.00 |
| | 200 | 10801.79 | 10802.02 | 1522.10 | 0 | 10 | 16.87 | 123.31 | 29.66 | 36.38 | 0 | 6.39 | 9.52 |
| Average | | 6235.78 | 7646.24 | 809.64 | 83(sum) | 99(sum) | 3.57 | 12.78 | 5.45 | 11.15 | 67(sum) | 2.17 | 6.71 |

**Table B-11:** Performance of the Solution Approaches for the Non-Resumable Jobs Case When $T_W$ is 100 and $T_U$ is 20

| | | MILP | | | | | | | Heuristic-NR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 1.48 | 1.70 | 1.48 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.56 | 1.70 | 1.56 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 1088.74 | 10802.14 | 2.05 | 9 | 1 | 0.24 | 2.36 | 0.05 | 0.06 | 9 | 0.24 | 2.36 |
| | 100 | 6485.56 | 10801.83 | 3.49 | 4 | 6 | 0.37 | 1.23 | 0.38 | 0.97 | 3 | 0.44 | 1.65 |
| | 200 | 10802.85 | 10810.47 | 2253.73 | 0 | 9 | 0.41 | 0.84 | 2.93 | 7.02 | 0 | 0.33 | 1.09 |
| 3 | 10 | 1.60 | 1.88 | 1.60 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.66 | 1.97 | 1.66 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 1102.68 | 10802.33 | 3.27 | 9 | 1 | 0.15 | 1.49 | 0.09 | 0.13 | 5 | 0.37 | 1.49 |
| | 100 | 6482.90 | 10802.00 | 3.18 | 4 | 6 | 0.46 | 1.23 | 0.47 | 0.84 | 3 | 0.49 | 1.54 |
| | 200 | 10801.99 | 10802.13 | 594.30 | 0 | 10 | 0.44 | 0.87 | 3.72 | 7.00 | 0 | 0.44 | 0.67 |
| 5 | 20 | 1.65 | 1.89 | 1.65 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 9 | 0.36 | 3.64 |
| | 50 | 23.75 | 115.86 | 8.85 | 10 | 0 | 0.00 | 0.00 | 0.12 | 0.17 | 5 | 0.47 | 1.95 |
| | 100 | 9799.33 | 10801.80 | 521.80 | 1 | 9 | 1.90 | 3.60 | 0.89 | 1.36 | 0 | 2.35 | 5.04 |
| | 200 | 10801.79 | 10801.97 | 1102.40 | 0 | 10 | 1.64 | 3.97 | 6.38 | 7.91 | 0 | 1.80 | 4.10 |
| 10 | 50 | 5.50 | 39.39 | 3.12 | 10 | 0 | 0.00 | 0.00 | 0.19 | 0.25 | 6 | 1.35 | 5.00 |
| | 100 | 1241.83 | 3974.25 | 914.79 | 10 | 0 | 0.00 | 0.00 | 1.50 | 1.89 | 0 | 1.99 | 5.26 |
| | 200 | 9722.77 | 10801.95 | 10.53 | 1 | 9 | 3.22 | 7.26 | 13.51 | 18.03 | 0 | 3.13 | 6.02 |
| 20 | 100 | 37.94 | 308.51 | 33.98 | 10 | 0 | 0.00 | 0.00 | 4.60 | 5.94 | 3 | 3.45 | 8.70 |
| | 200 | 231.73 | 1665.94 | 46.87 | 10 | 0 | 0.00 | 0.00 | 28.65 | 41.00 | 2 | 2.54 | 4.65 |
| Average | | 3612.49 | 5438.93 | 290.02 | 128(sum) | 61(sum) | 0.46 | 1.20 | 3.34 | 4.87 | 85(sum) | 1.04 | 2.80 |

**Table B-12:** Performance of the Solution Approaches for the Non-Resumable Jobs Case When $T_W$ is 100 and $T_U$ is 50

| | | MILP | | | | | | | Heuristic-NR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 1.55 | 1.78 | 1.55 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.54 | 1.64 | 1.54 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 27.73 | 250.39 | 2.87 | 10 | 0 | 0.00 | 0.00 | 0.05 | 0.06 | 10 | 0.00 | 0.00 |
| | 100 | 6483.15 | 10801.78 | 5.24 | 4 | 6 | 0.32 | 0.99 | 0.54 | 1.70 | 4 | 0.35 | 1.32 |
| | 200 | 8412.26 | 10801.98 | 549.16 | 2 | 7 | 1.00 | 2.42 | 4.97 | 16.50 | 0 | 0.46 | 1.83 |
| 3 | 10 | 1.49 | 1.69 | 1.49 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.64 | 1.80 | 1.64 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 1214.33 | 10801.70 | 3.06 | 9 | 1 | 0.08 | 0.77 | 0.10 | 0.23 | 5 | 0.26 | 0.77 |
| | 100 | 8643.74 | 10802.11 | 35.78 | 2 | 8 | 1.01 | 6.79 | 0.57 | 1.78 | 1 | 0.46 | 1.25 |
| | 200 | 10801.87 | 10802.14 | 90.05 | 0 | 10 | 0.62 | 3.52 | 5.14 | 16.86 | 0 | 0.36 | 0.54 |
| 5 | 20 | 1.57 | 1.70 | 1.57 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 9 | 0.36 | 3.64 |
| | 50 | 12.43 | 36.41 | 5.24 | 10 | 0 | 0.00 | 0.00 | 0.12 | 0.17 | 5 | 0.39 | 1.63 |
| | 100 | 10801.61 | 10801.74 | 1012.23 | 0 | 10 | 1.89 | 3.85 | 0.89 | 1.36 | 0 | 1.98 | 4.14 |
| | 200 | 10801.74 | 10801.89 | 1921.85 | 0 | 10 | 3.04 | 9.80 | 7.29 | 13.63 | 0 | 2.32 | 7.57 |
| 10 | 50 | 11.42 | 98.72 | 10.10 | 10 | 0 | 0.00 | 0.00 | 0.19 | 0.25 | 6 | 1.35 | 5.00 |
| | 100 | 971.75 | 3110.92 | 904.83 | 10 | 0 | 0.00 | 0.00 | 1.53 | 1.91 | 1 | 1.87 | 5.26 |
| | 200 | 9722.14 | 10801.97 | 1426.22 | 1 | 9 | 1.90 | 5.10 | 13.47 | 18.03 | 0 | 2.48 | 5.10 |
| 20 | 100 | 38.28 | 308.83 | 34.26 | 10 | 0 | 0.00 | 0.00 | 4.60 | 5.92 | 3 | 3.45 | 8.70 |
| | 200 | 232.12 | 1667.53 | 46.37 | 10 | 0 | 0.00 | 0.00 | 28.66 | 41.02 | 2 | 2.54 | 4.65 |
| Average | | 3588.54 | 4836.67 | 318.69 | 128(sum) | 61(sum) | 0.52 | 1.75 | 3.59 | 6.29 | 86(sum) | 0.98 | 2.71 |

**Table B-13:** Performance of the Solution Approaches for the Non-Resumable Jobs Case When $T_W$ is 100 and $T_U$ is 100

| | | MILP | | | | | | | Heuristic-NR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 1.60 | 1.89 | 1.60 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.81 | 2.72 | 1.81 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 1082.48 | 10802.31 | 2.46 | 9 | 1 | 0.08 | 0.76 | 0.05 | 0.06 | 9 | 0.08 | 0.76 |
| | 100 | 7564.78 | 10802.09 | 6.34 | 3 | 7 | 0.83 | 5.80 | 0.50 | 1.33 | 3 | 0.25 | 0.99 |
| | 200 | 9062.59 | 10802.11 | 303.77 | 2 | 8 | 0.65 | 2.46 | 4.97 | 16.48 | 0 | 0.53 | 2.61 |
| 3 | 10 | 1.46 | 1.69 | 1.46 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.66 | 2.00 | 1.66 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 70.80 | 422.17 | 6.82 | 10 | 0 | 0.00 | 0.00 | 0.10 | 0.23 | 6 | 0.14 | 0.48 |
| | 100 | 9721.83 | 10801.89 | 216.15 | 1 | 9 | 0.39 | 1.05 | 0.57 | 1.78 | 0 | 0.46 | 0.95 |
| | 200 | 10801.83 | 10801.92 | 2412.27 | 0 | 9 | 0.53 | 2.56 | 5.14 | 16.88 | 0 | 0.29 | 0.41 |
| 5 | 20 | 1.65 | 1.89 | 1.65 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 9 | 0.36 | 3.64 |
| | 50 | 18.83 | 66.17 | 8.37 | 10 | 0 | 0.00 | 0.00 | 0.11 | 0.16 | 5 | 0.31 | 1.28 |
| | 100 | 8874.52 | 10801.92 | 659.34 | 2 | 8 | 1.29 | 3.20 | 0.89 | 1.34 | 0 | 1.49 | 3.20 |
| | 200 | 10801.89 | 10802.09 | 787.82 | 0 | 10 | 2.96 | 12.11 | 7.28 | 13.61 | 0 | 2.88 | 11.44 |
| 10 | 50 | 4.11 | 26.31 | 3.09 | 10 | 0 | 0.00 | 0.00 | 0.20 | 0.25 | 6 | 1.35 | 5.00 |
| | 100 | 706.56 | 2091.84 | 507.97 | 10 | 0 | 0.00 | 0.00 | 1.53 | 1.89 | 1 | 1.87 | 5.26 |
| | 200 | 10801.78 | 10801.86 | 608.90 | 0 | 10 | 2.26 | 5.02 | 13.51 | 18.02 | 0 | 2.05 | 4.07 |
| 20 | 100 | 38.27 | 308.56 | 34.28 | 10 | 0 | 0.00 | 0.00 | 4.60 | 5.92 | 3 | 3.45 | 8.70 |
| | 200 | 231.86 | 1664.69 | 47.17 | 10 | 0 | 0.00 | 0.00 | 28.65 | 41.02 | 2 | 2.54 | 4.65 |
| Average | | 3673.17 | 4789.80 | 295.42 | 127(sum) | 62(sum) | 0.47 | 1.73 | 3.59 | 6.26 | 84(sum) | 0.95 | 2.81 |

**Table B-14:** Performance of the Solution Approaches for the Non-Resumable Jobs Case When $T_W$ is 150 and $T_U$ is 30

| | | MILP | | | | | | | Heuristic-NR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 1.58 | 1.84 | 1.58 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.57 | 1.73 | 1.57 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 2.45 | 8.69 | 2.45 | 10 | 0 | 0.00 | 0.00 | 0.04 | 0.06 | 10 | 0.00 | 0.00 |
| | 100 | 2164.95 | 10801.89 | 3.70 | 8 | 2 | 0.07 | 0.41 | 0.34 | 0.67 | 5 | 0.13 | 0.67 |
| | 200 | 7574.52 | 10801.91 | 11.30 | 3 | 7 | 0.22 | 0.76 | 2.76 | 8.59 | 0 | 0.25 | 0.88 |
| 3 | 10 | 1.45 | 1.72 | 1.45 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.63 | 1.75 | 1.63 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 3.09 | 8.64 | 3.09 | 10 | 0 | 0.00 | 0.00 | 0.08 | 0.14 | 6 | 0.23 | 0.78 |
| | 100 | 6488.40 | 10801.73 | 265.03 | 4 | 6 | 0.40 | 1.08 | 0.48 | 1.19 | 3 | 0.55 | 1.25 |
| | 200 | 10801.78 | 10801.89 | 51.10 | 0 | 10 | 0.60 | 2.48 | 4.24 | 10.70 | 0 | 0.46 | 0.67 |
| 5 | 20 | 1.55 | 1.66 | 1.55 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 9 | 0.36 | 3.64 |
| | 50 | 4.10 | 9.92 | 4.10 | 10 | 0 | 0.00 | 0.00 | 0.08 | 0.11 | 5 | 0.54 | 2.24 |
| | 100 | 6554.78 | 10801.95 | 1136.73 | 5 | 5 | 0.97 | 3.42 | 0.85 | 1.42 | 0 | 1.47 | 4.18 |
| | 200 | 10801.74 | 10801.86 | 2148.64 | 0 | 10 | 1.16 | 2.00 | 5.84 | 7.31 | 0 | 1.19 | 1.95 |
| 10 | 50 | 1.64 | 1.84 | 1.64 | 10 | 0 | 0.00 | 0.00 | 0.20 | 0.25 | 6 | 1.35 | 5.00 |
| | 100 | 166.49 | 920.34 | 122.31 | 10 | 0 | 0.00 | 0.00 | 1.50 | 2.00 | 1 | 1.98 | 5.26 |
| | 200 | 6547.84 | 10801.83 | 277.29 | 4 | 6 | 6.69 | 21.32 | 15.96 | 20.70 | 0 | 7.96 | 23.03 |
| 20 | 100 | 8.07 | 30.08 | 8.07 | 10 | 0 | 0.00 | 0.00 | 4.60 | 5.94 | 3 | 3.45 | 8.70 |
| | 200 | 111.71 | 461.64 | 41.29 | 10 | 0 | 0.00 | 0.00 | 28.65 | 41.00 | 2 | 2.54 | 4.65 |
| Average | | 2696.81 | 4055.94 | 214.97 | 144(sum) | 46(sum) | 0.53 | 1.66 | 3.45 | 5.27 | 90(sum) | 1.18 | 3.31 |

**Table B-15:** Performance of the Solution Approaches for the Non-Resumable Jobs Case When $T_W$ is 150 and $T_U$ is 75

| | | MILP | | | | | | | Heuristic-NR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 1.52 | 1.69 | 1.52 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.54 | 1.62 | 1.54 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 2.35 | 6.23 | 2.35 | 10 | 0 | 0.00 | 0.00 | 0.04 | 0.06 | 10 | 0.00 | 0.00 |
| | 100 | 3244.34 | 10801.77 | 56.10 | 7 | 3 | 0.05 | 0.41 | 0.38 | 0.98 | 5 | 0.13 | 0.75 |
| | 200 | 9723.75 | 10801.95 | 44.16 | 1 | 9 | 0.25 | 0.81 | 3.45 | 13.80 | 0 | 0.30 | 0.94 |
| 3 | 10 | 1.64 | 2.01 | 1.64 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.76 | 1.92 | 1.76 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 3.00 | 4.44 | 3.00 | 10 | 0 | 0.00 | 0.00 | 0.09 | 0.20 | 6 | 0.19 | 0.63 |
| | 100 | 7562.09 | 10802.23 | 1563.37 | 3 | 7 | 0.33 | 1.16 | 0.52 | 1.66 | 2 | 0.44 | 1.29 |
| | 200 | 10801.90 | 10802.36 | 340.60 | 0 | 10 | 0.90 | 5.38 | 4.98 | 15.70 | 0 | 0.36 | 0.54 |
| 5 | 20 | 1.53 | 1.70 | 1.53 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 9 | 0.36 | 3.64 |
| | 50 | 4.10 | 7.77 | 4.10 | 10 | 0 | 0.00 | 0.00 | 0.08 | 0.11 | 5 | 0.54 | 2.24 |
| | 100 | 8249.16 | 10801.78 | 657.01 | 3 | 7 | 1.03 | 3.25 | 0.80 | 1.25 | 0 | 1.34 | 3.57 |
| | 200 | 10801.67 | 10801.80 | 24.22 | 0 | 10 | 0.91 | 1.46 | 5.82 | 7.31 | 0 | 0.98 | 1.79 |
| 10 | 50 | 3.40 | 17.27 | 2.83 | 10 | 0 | 0.00 | 0.00 | 0.19 | 0.25 | 6 | 1.35 | 5.00 |
| | 100 | 317.11 | 3023.12 | 300.72 | 10 | 0 | 0.00 | 0.00 | 1.50 | 2.00 | 1 | 1.98 | 5.26 |
| | 200 | 5651.88 | 10802.27 | 2681.59 | 6 | 4 | 15.28 | 50.90 | 17.75 | 23.97 | 0 | 16.55 | 52.22 |
| 20 | 100 | 8.05 | 29.80 | 8.05 | 10 | 0 | 0.00 | 0.00 | 4.60 | 5.94 | 3 | 3.45 | 8.70 |
| | 200 | 111.71 | 462.69 | 41.28 | 10 | 0 | 0.00 | 0.00 | 28.65 | 41.00 | 2 | 2.54 | 4.65 |
| Average | | 2973.29 | 4167.07 | 301.97 | 140(sum) | 50(sum) | 0.99 | 3.33 | 3.63 | 6.02 | 89(sum) | 1.61 | 4.80 |

**Table B-16:** Performance of the Solution Approaches for the Non-Resumable Jobs Case When $T_W$ is 150 and $T_U$ is 150

| | | MILP | | | | | | | Heuristic-NR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 1.48 | 1.61 | 1.48 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.54 | 1.61 | 1.54 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 1082.65 | 10801.84 | 2.68 | 9 | 1 | 0.08 | 0.79 | 0.04 | 0.06 | 9 | 0.08 | 0.79 |
| | 100 | 3244.47 | 10801.89 | 3.61 | 7 | 3 | 0.09 | 0.78 | 0.39 | 0.98 | 5 | 0.15 | 1.04 |
| | 200 | 7625.48 | 10801.91 | 355.79 | 3 | 7 | 0.60 | 4.30 | 3.45 | 13.80 | 0 | 0.22 | 0.71 |
| 3 | 10 | 1.53 | 1.66 | 1.53 | 10 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 10 | 0.00 | 0.00 |
| | 20 | 1.56 | 1.62 | 1.56 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 10 | 0.00 | 0.00 |
| | 50 | 454.03 | 4509.53 | 2.35 | 10 | 0 | 0.00 | 0.00 | 0.09 | 0.20 | 6 | 0.15 | 0.48 |
| | 100 | 7562.35 | 10801.92 | 134.38 | 3 | 7 | 0.26 | 0.90 | 0.52 | 1.66 | 1 | 0.36 | 1.00 |
| | 200 | 9724.31 | 10802.13 | 365.45 | 1 | 9 | 1.13 | 8.19 | 4.98 | 15.77 | 0 | 0.24 | 0.41 |
| 5 | 20 | 1.65 | 1.86 | 1.65 | 10 | 0 | 0.00 | 0.00 | 0.01 | 0.02 | 9 | 0.36 | 3.64 |
| | 50 | 3.20 | 7.77 | 3.20 | 10 | 0 | 0.00 | 0.00 | 0.08 | 0.11 | 5 | 0.54 | 2.24 |
| | 100 | 8294.73 | 10802.08 | 1776.29 | 4 | 6 | 0.85 | 2.48 | 0.81 | 1.27 | 0 | 1.12 | 2.87 |
| | 200 | 10801.85 | 10802.03 | 1094.35 | 0 | 10 | 0.62 | 1.26 | 5.81 | 7.30 | 0 | 0.68 | 1.39 |
| 10 | 50 | 1.79 | 2.30 | 1.79 | 10 | 0 | 0.00 | 0.00 | 0.20 | 0.25 | 6 | 1.35 | 5.00 |
| | 100 | 223.00 | 1379.61 | 199.46 | 10 | 0 | 0.00 | 0.00 | 1.49 | 2.00 | 1 | 1.98 | 5.26 |
| | 200 | 6289.58 | 10801.97 | 1895.21 | 5 | 5 | 39.78 | 103.38 | 17.66 | 23.98 | 0 | 30.90 | 101.32 |
| 20 | 100 | 7.68 | 29.48 | 7.68 | 10 | 0 | 0.00 | 0.00 | 4.60 | 5.94 | 3 | 3.45 | 8.70 |
| | 200 | 111.15 | 460.64 | 41.16 | 10 | 0 | 0.00 | 0.00 | 28.65 | 41.02 | 2 | 2.54 | 4.65 |
| Average | | 2917.58 | 4884.92 | 310.06 | 142(sum) | 48(sum) | 2.28 | 6.43 | 3.62 | 6.02 | 87(sum) | 2.32 | 7.34 |

# APPENDIX C Performance of Solution Approaches for Resumable Jobs

**Table C-1:** Performance of the solution approaches for the resumable jobs case when $T_W$ is 10 and $T_U$ is 5

| | | MILP | | | | | | | Heuristic-R | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 3.15 | 3.34 | 3.15 | 10 | 0 | 0.00 | 0.00 | 3.09 | 3.61 | 10 | 0.00 | 0.00 |
| | 20 | 3.19 | 3.46 | 3.19 | 10 | 0 | 0.00 | 0.00 | 3.53 | 4.00 | 10 | 0.00 | 0.00 |
| | 50 | 3.39 | 5.20 | 3.39 | 10 | 0 | 0.00 | 0.00 | 3.54 | 4.02 | 10 | 0.00 | 0.00 |
| | 100 | 3.18 | 3.31 | 3.18 | 10 | 0 | 0.00 | 0.00 | 3.87 | 4.39 | 10 | 0.00 | 0.00 |
| | 200 | 3.21 | 3.33 | 3.21 | 10 | 0 | 0.00 | 0.00 | 4.79 | 6.83 | 9 | 0.02 | 0.15 |
| 3 | 10 | 3.07 | 3.18 | 3.07 | 10 | 0 | 0.00 | 0.00 | 3.41 | 4.06 | 10 | 0.00 | 0.00 |
| | 20 | 3.16 | 3.29 | 3.16 | 10 | 0 | 0.00 | 0.00 | 3.49 | 3.77 | 10 | 0.00 | 0.00 |
| | 50 | 3.17 | 3.26 | 3.17 | 10 | 0 | 0.00 | 0.00 | 3.60 | 4.08 | 10 | 0.00 | 0.00 |
| | 100 | 3.30 | 3.56 | 3.30 | 10 | 0 | 0.00 | 0.00 | 3.93 | 4.73 | 7 | 0.15 | 0.51 |
| | 200 | 81.19 | 737.29 | 4.07 | 10 | 0 | 0.00 | 0.00 | 6.61 | 11.48 | 9 | 0.03 | 0.28 |
| 5 | 20 | 3.17 | 3.29 | 3.17 | 10 | 0 | 0.00 | 0.00 | 3.59 | 4.28 | 10 | 0.00 | 0.00 |
| | 50 | 3.71 | 5.58 | 3.71 | 10 | 0 | 0.00 | 0.00 | 3.65 | 3.96 | 7 | 0.61 | 2.13 |
| | 100 | 63.37 | 231.29 | 3.51 | 10 | 0 | 0.00 | 0.00 | 4.16 | 4.87 | 8 | 0.23 | 1.27 |
| | 200 | 3557.84 | 10803.76 | 3.48 | 7 | 3 | 0.11 | 0.48 | 8.36 | 12.59 | 2 | 0.41 | 0.60 |
| 10 | 50 | 318.37 | 1499.19 | 3.58 | 10 | 0 | 0.00 | 0.00 | 3.79 | 4.29 | 9 | 6.00 | 60.00 |
| | 100 | 2562.49 | 10803.14 | 3.23 | 8 | 2 | 0.68 | 3.54 | 4.73 | 5.77 | 5 | 1.88 | 4.35 |
| | 200 | 4649.09 | 10803.47 | 3.60 | 6 | 4 | 0.42 | 1.23 | 13.19 | 22.64 | 2 | 1.17 | 2.00 |
| 20 | 100 | 6016.48 | 10803.23 | 4.14 | 5 | 5 | 2.67 | 10.14 | 5.84 | 7.95 | 5 | 2.67 | 10.14 |
| | 200 | 8686.80 | 10803.43 | 3.83 | 3 | 7 | 1.88 | 4.40 | 43.78 | 68.01 | 0 | 3.55 | 5.56 |
| Average | | 1366.91 | 2975.03 | 3.43 | 169(sum) | 21(sum) | 0.30 | 1.04 | 6.89 | 9.75 | 143(sum) | 0.88 | 4.58 |

**Table C-2:** Performance of the Solution Approaches for the Resumable Jobs Case When $T_W$ is 10 and $T_U$ is 10

| | | MILP | | | | | | | Heuristic-R | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 3.18 | 3.39 | 3.18 | 10 | 0 | 0.00 | 0.00 | 3.06 | 3.63 | 10 | 0.00 | 0.00 |
| | 20 | 3.19 | 3.34 | 3.19 | 10 | 0 | 0.00 | 0.00 | 3.48 | 3.85 | 10 | 0.00 | 0.00 |
| | 50 | 3.20 | 3.36 | 3.20 | 10 | 0 | 0.00 | 0.00 | 3.47 | 4.14 | 10 | 0.00 | 0.00 |
| | 100 | 3.21 | 3.36 | 3.21 | 10 | 0 | 0.00 | 0.00 | 3.82 | 4.93 | 10 | 0.00 | 0.00 |
| | 200 | 3.24 | 3.37 | 3.24 | 10 | 0 | 0.00 | 0.00 | 5.47 | 9.17 | 8 | 0.02 | 0.12 |
| 3 | 10 | 3.11 | 3.29 | 3.11 | 10 | 0 | 0.00 | 0.00 | 3.33 | 3.84 | 10 | 0.00 | 0.00 |
| | 20 | 3.15 | 3.25 | 3.15 | 10 | 0 | 0.00 | 0.00 | 3.50 | 3.77 | 10 | 0.00 | 0.00 |
| | 50 | 3.21 | 3.33 | 3.21 | 10 | 0 | 0.00 | 0.00 | 3.44 | 3.86 | 10 | 0.00 | 0.00 |
| | 100 | 3.29 | 3.64 | 3.29 | 10 | 0 | 0.00 | 0.00 | 3.97 | 4.70 | 7 | 0.11 | 0.38 |
| | 200 | 81.76 | 765.00 | 3.42 | 10 | 0 | 0.00 | 0.00 | 8.94 | 19.54 | 9 | 0.02 | 0.21 |
| 5 | 20 | 3.19 | 3.35 | 3.19 | 10 | 0 | 0.00 | 0.00 | 3.35 | 3.99 | 10 | 0.00 | 0.00 |
| | 50 | 3.84 | 5.78 | 3.84 | 10 | 0 | 0.00 | 0.00 | 3.60 | 3.95 | 6 | 0.69 | 2.22 |
| | 100 | 36.96 | 141.03 | 3.47 | 10 | 0 | 0.00 | 0.00 | 4.24 | 5.40 | 8 | 0.18 | 0.96 |
| | 200 | 4085.72 | 10804.27 | 4.00 | 7 | 3 | 0.08 | 0.36 | 10.93 | 21.17 | 3 | 0.26 | 0.45 |
| 10 | 50 | 223.68 | 1193.21 | 3.80 | 10 | 0 | 0.00 | 0.00 | 3.73 | 4.26 | 9 | 11.00 | 110.00 |
| | 100 | 2560.93 | 10803.63 | 80.88 | 8 | 2 | 0.57 | 2.96 | 4.83 | 6.14 | 5 | 1.52 | 3.57 |
| | 200 | 4426.68 | 10803.75 | 14.14 | 6 | 4 | 0.32 | 0.91 | 15.14 | 24.23 | 2 | 0.90 | 1.54 |
| 20 | 100 | 6525.37 | 10803.51 | 3.65 | 5 | 5 | 2.42 | 9.63 | 5.64 | 7.69 | 5 | 2.42 | 9.63 |
| | 200 | 8643.97 | 10803.78 | 4.19 | 2 | 8 | 1.50 | 3.35 | 48.28 | 74.86 | 0 | 2.80 | 4.48 |
| Average | | 1401.10 | 2955.66 | 8.07 | 168(sum) | 22(sum) | 0.26 | 0.91 | 7.49 | 11.22 | 142(sum) | 1.05 | 7.03 |

**Table C-3:** Performance of the Solution Approaches for the Resumable Jobs Case When $T_W$ is 20 and $T_U$ is 4

| | | MILP | | | | | | | Heuristic-R | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 3.21 | 3.52 | 3.21 | 10 | 0 | 0.00 | 0.00 | 3.07 | 3.60 | 10 | 0.00 | 0.00 |
| | 20 | 3.35 | 3.47 | 3.35 | 10 | 0 | 0.00 | 0.00 | 3.46 | 3.75 | 10 | 0.00 | 0.00 |
| | 50 | 3.35 | 3.44 | 3.35 | 10 | 0 | 0.00 | 0.00 | 3.56 | 3.85 | 9 | 0.08 | 0.77 |
| | 100 | 3.26 | 3.45 | 3.26 | 10 | 0 | 0.00 | 0.00 | 3.70 | 4.15 | 10 | 0.00 | 0.00 |
| | 200 | 3.37 | 3.64 | 3.37 | 10 | 0 | 0.00 | 0.00 | 4.43 | 5.60 | 10 | 0.00 | 0.00 |
| 3 | 10 | 3.25 | 3.32 | 3.25 | 10 | 0 | 0.00 | 0.00 | 3.46 | 3.84 | 10 | 0.00 | 0.00 |
| | 20 | 3.25 | 3.46 | 3.25 | 10 | 0 | 0.00 | 0.00 | 3.54 | 3.96 | 10 | 0.00 | 0.00 |
| | 50 | 3.20 | 3.35 | 3.20 | 10 | 0 | 0.00 | 0.00 | 3.56 | 3.94 | 9 | 0.13 | 1.27 |
| | 100 | 3.31 | 3.59 | 3.31 | 10 | 0 | 0.00 | 0.00 | 3.81 | 4.54 | 7 | 0.19 | 0.64 |
| | 200 | 98.42 | 901.61 | 3.90 | 10 | 0 | 0.00 | 0.00 | 6.17 | 9.01 | 9 | 0.03 | 0.34 |
| 5 | 20 | 3.16 | 3.30 | 3.16 | 10 | 0 | 0.00 | 0.00 | 3.50 | 3.87 | 10 | 0.00 | 0.00 |
| | 50 | 3.79 | 5.49 | 3.79 | 10 | 0 | 0.00 | 0.00 | 3.62 | 3.96 | 7 | 0.87 | 3.45 |
| | 100 | 73.55 | 266.63 | 5.08 | 10 | 0 | 0.00 | 0.00 | 4.06 | 4.56 | 8 | 0.29 | 1.61 |
| | 200 | 3388.82 | 10803.98 | 4.38 | 7 | 3 | 0.13 | 0.59 | 7.76 | 12.66 | 3 | 0.43 | 0.76 |
| 10 | 50 | 60.68 | 270.03 | 7.31 | 10 | 0 | 0.00 | 0.00 | 3.67 | 4.04 | 9 | 1.00 | 10.00 |
| | 100 | 2597.67 | 10803.44 | 173.26 | 8 | 2 | 0.89 | 4.55 | 4.53 | 5.46 | 7 | 1.29 | 4.55 |
| | 200 | 5426.34 | 10803.64 | 26.25 | 5 | 5 | 0.54 | 1.66 | 12.54 | 18.00 | 2 | 1.49 | 2.56 |
| 20 | 100 | 5993.74 | 10803.54 | 4.07 | 6 | 4 | 2.24 | 9.57 | 5.55 | 7.84 | 6 | 2.24 | 9.57 |
| | 200 | 8699.16 | 10804.12 | 4.09 | 2 | 8 | 2.59 | 5.90 | 14.02 | 25.92 | 2 | 2.59 | 5.90 |
| Average | | 1388.15 | 2920.90 | 13.94 | 168(sum) | 22(sum) | 0.34 | 1.17 | 5.16 | 6.98 | 148(sum) | 0.56 | 2.18 |

**Table C-4:** Performance of the Solution Approaches for the Resumable Jobs Case When $T_W$ is 20 and $T_U$ is 10

| | | MILP | | | | | | | Heuristic-R | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 3.12 | 3.36 | 3.12 | 10 | 0 | 0.00 | 0.00 | 3.07 | 3.50 | 10 | 0.00 | 0.00 |
| | 20 | 3.20 | 3.38 | 3.20 | 10 | 0 | 0.00 | 0.00 | 3.49 | 3.75 | 10 | 0.00 | 0.00 |
| | 50 | 3.16 | 3.30 | 3.16 | 10 | 0 | 0.00 | 0.00 | 3.55 | 4.00 | 10 | 0.00 | 0.00 |
| | 100 | 3.19 | 3.30 | 3.19 | 10 | 0 | 0.00 | 0.00 | 3.72 | 4.27 | 10 | 0.00 | 0.00 |
| | 200 | 3.19 | 3.26 | 3.19 | 10 | 0 | 0.00 | 0.00 | 4.54 | 6.90 | 8 | 0.03 | 0.16 |
| 3 | 10 | 3.05 | 3.15 | 3.05 | 10 | 0 | 0.00 | 0.00 | 3.42 | 3.90 | 10 | 0.00 | 0.00 |
| | 20 | 3.14 | 3.37 | 3.14 | 10 | 0 | 0.00 | 0.00 | 3.44 | 3.68 | 10 | 0.00 | 0.00 |
| | 50 | 3.20 | 3.27 | 3.20 | 10 | 0 | 0.00 | 0.00 | 3.65 | 3.81 | 9 | 0.10 | 1.03 |
| | 100 | 3.24 | 3.37 | 3.24 | 10 | 0 | 0.00 | 0.00 | 3.96 | 4.73 | 7 | 0.15 | 0.52 |
| | 200 | 95.49 | 885.68 | 3.90 | 10 | 0 | 0.00 | 0.00 | 8.84 | 19.20 | 9 | 0.03 | 0.28 |
| 5 | 20 | 3.12 | 3.27 | 3.12 | 10 | 0 | 0.00 | 0.00 | 3.56 | 4.27 | 10 | 0.00 | 0.00 |
| | 50 | 3.82 | 6.46 | 3.82 | 10 | 0 | 0.00 | 0.00 | 3.69 | 4.06 | 6 | 0.96 | 2.86 |
| | 100 | 51.58 | 161.80 | 3.63 | 10 | 0 | 0.00 | 0.00 | 4.23 | 4.91 | 8 | 0.24 | 1.35 |
| | 200 | 3569.06 | 10804.02 | 4.38 | 7 | 3 | 0.11 | 0.50 | 10.01 | 18.47 | 2 | 0.41 | 0.62 |
| 10 | 50 | 53.12 | 246.60 | 7.01 | 10 | 0 | 0.00 | 0.00 | 3.63 | 4.02 | 9 | 1.00 | 10.00 |
| | 100 | 2501.87 | 10803.31 | 3.92 | 8 | 2 | 0.88 | 4.55 | 4.86 | 6.85 | 6 | 1.75 | 5.56 |
| | 200 | 6412.87 | 10803.61 | 1012.80 | 5 | 5 | 0.45 | 1.30 | 13.58 | 24.04 | 2 | 1.27 | 2.22 |
| 20 | 100 | 6621.05 | 10805.10 | 3.42 | 5 | 5 | 2.67 | 10.15 | 5.62 | 7.58 | 5 | 2.67 | 10.15 |
| | 200 | 9723.84 | 10804.60 | 3.67 | 2 | 8 | 2.51 | 5.90 | 13.67 | 24.64 | 2 | 2.51 | 5.90 |
| Average | | 1529.70 | 2913.38 | 56.74 | 167(sum) | 23(sum) | 0.35 | 1.18 | 5.50 | 8.24 | 143(sum) | 0.59 | 2.14 |

**Table C-5:** Performance of the Solution Approaches for the Resumable Jobs Case When $T_W$ is 20 and $T_U$ is 20

| | | MILP | | | | | | | Heuristic-R | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 2.73 | 2.99 | 2.73 | 10 | 0 | 0.00 | 0.00 | 3.16 | 3.45 | 10 | 0.00 | 0.00 |
| | 20 | 2.78 | 2.87 | 2.78 | 10 | 0 | 0.00 | 0.00 | 3.51 | 4.00 | 10 | 0.00 | 0.00 |
| | 50 | 2.77 | 2.83 | 2.77 | 10 | 0 | 0.00 | 0.00 | 3.52 | 3.75 | 10 | 0.00 | 0.00 |
| | 100 | 2.83 | 2.97 | 2.83 | 10 | 0 | 0.00 | 0.00 | 3.76 | 4.47 | 10 | 0.00 | 0.00 |
| | 200 | 2.94 | 3.70 | 2.94 | 10 | 0 | 0.00 | 0.00 | 4.78 | 7.82 | 8 | 0.02 | 0.12 |
| 3 | 10 | 2.68 | 2.80 | 2.68 | 10 | 0 | 0.00 | 0.00 | 3.45 | 3.90 | 10 | 0.00 | 0.00 |
| | 20 | 2.74 | 2.85 | 2.74 | 10 | 0 | 0.00 | 0.00 | 3.55 | 3.75 | 10 | 0.00 | 0.00 |
| | 50 | 2.84 | 2.93 | 2.84 | 10 | 0 | 0.00 | 0.00 | 3.58 | 4.01 | 9 | 0.08 | 0.79 |
| | 100 | 2.91 | 3.09 | 2.91 | 10 | 0 | 0.00 | 0.00 | 3.99 | 4.81 | 7 | 0.11 | 0.40 |
| | 200 | 99.45 | 918.51 | 2.97 | 10 | 0 | 0.00 | 0.00 | 9.09 | 19.88 | 9 | 0.02 | 0.21 |
| 5 | 20 | 2.77 | 2.88 | 2.77 | 10 | 0 | 0.00 | 0.00 | 3.44 | 4.11 | 10 | 0.00 | 0.00 |
| | 50 | 3.41 | 5.00 | 3.41 | 10 | 0 | 0.00 | 0.00 | 3.69 | 4.43 | 7 | 0.55 | 1.92 |
| | 100 | 41.23 | 185.83 | 3.75 | 10 | 0 | 0.00 | 0.00 | 4.21 | 5.04 | 8 | 0.19 | 1.06 |
| | 200 | 2937.36 | 10803.63 | 3.33 | 8 | 2 | 0.07 | 0.38 | 11.13 | 21.78 | 4 | 0.25 | 0.47 |
| 10 | 50 | 45.55 | 219.32 | 4.90 | 10 | 0 | 0.00 | 0.00 | 3.71 | 3.99 | 9 | 1.00 | 10.00 |
| | 100 | 2731.73 | 10803.69 | 18.68 | 8 | 2 | 0.88 | 4.66 | 4.84 | 6.48 | 6 | 1.67 | 5.56 |
| | 200 | 4403.81 | 10804.19 | 3.75 | 6 | 4 | 0.37 | 1.08 | 15.88 | 29.89 | 2 | 1.02 | 1.82 |
| 20 | 100 | 6968.22 | 10803.61 | 3.87 | 4 | 6 | 2.81 | 10.15 | 5.75 | 7.77 | 4 | 2.81 | 10.15 |
| | 200 | 8755.99 | 10803.76 | 3.56 | 3 | 7 | 2.65 | 6.06 | 13.68 | 27.89 | 2 | 3.42 | 7.69 |
| Average | | 1369.20 | 2914.60 | 4.01 | 169(sum) | 21(sum) | 0.36 | 1.18 | 5.72 | 9.01 | 145(sum) | 0.59 | 2.11 |

**Table C-6:** Performance of the Solution Approaches for the Resumable Jobs Case When $T_W$ is 30 and $T_U$ is 6

| | | MILP | | | | | | | Heuristic-R | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 3.11 | 3.17 | 3.11 | 10 | 0 | 0.00 | 0.00 | 2.74 | 3.12 | 10 | 0.00 | 0.00 |
| | 20 | 3.14 | 3.24 | 3.14 | 10 | 0 | 0.00 | 0.00 | 2.72 | 2.85 | 10 | 0.00 | 0.00 |
| | 50 | 3.13 | 3.27 | 3.13 | 10 | 0 | 0.00 | 0.00 | 2.76 | 3.25 | 9 | 0.08 | 0.78 |
| | 100 | 3.20 | 3.31 | 3.20 | 10 | 0 | 0.00 | 0.00 | 2.98 | 3.55 | 10 | 0.00 | 0.00 |
| | 200 | 3.20 | 3.36 | 3.20 | 10 | 0 | 0.00 | 0.00 | 3.71 | 5.39 | 8 | 0.04 | 0.19 |
| 3 | 10 | 3.02 | 3.10 | 3.02 | 10 | 0 | 0.00 | 0.00 | 2.65 | 2.90 | 10 | 0.00 | 0.00 |
| | 20 | 3.13 | 3.25 | 3.13 | 10 | 0 | 0.00 | 0.00 | 2.75 | 3.19 | 10 | 0.00 | 0.00 |
| | 50 | 3.21 | 3.27 | 3.21 | 10 | 0 | 0.00 | 0.00 | 2.79 | 3.23 | 9 | 0.13 | 1.27 |
| | 100 | 3.25 | 3.42 | 3.25 | 10 | 0 | 0.00 | 0.00 | 3.20 | 3.81 | 7 | 0.19 | 0.64 |
| | 200 | 45.76 | 375.90 | 4.07 | 10 | 0 | 0.00 | 0.00 | 4.67 | 6.01 | 9 | 0.03 | 0.34 |
| 5 | 20 | 3.12 | 3.18 | 3.12 | 10 | 0 | 0.00 | 0.00 | 2.75 | 3.24 | 10 | 0.00 | 0.00 |
| | 50 | 3.74 | 5.50 | 3.74 | 10 | 0 | 0.00 | 0.00 | 2.91 | 3.49 | 7 | 0.75 | 2.63 |
| | 100 | 54.86 | 248.11 | 3.71 | 10 | 0 | 0.00 | 0.00 | 3.40 | 3.90 | 8 | 0.30 | 1.67 |
| | 200 | 4200.87 | 10804.15 | 3.47 | 7 | 3 | 0.13 | 0.61 | 7.59 | 13.16 | 4 | 0.36 | 0.77 |
| 10 | 50 | 54.21 | 302.11 | 6.84 | 10 | 0 | 0.00 | 0.00 | 2.92 | 3.43 | 9 | 1.00 | 10.00 |
| | 100 | 2355.26 | 10803.44 | 3.46 | 8 | 2 | 0.87 | 4.54 | 3.68 | 4.43 | 6 | 1.91 | 5.56 |
| | 200 | 4533.41 | 10807.52 | 11.23 | 6 | 4 | 0.53 | 1.69 | 14.90 | 19.42 | 1 | 1.68 | 2.44 |
| 20 | 100 | 5528.68 | 10803.64 | 3.58 | 5 | 5 | 2.51 | 9.44 | 5.06 | 7.12 | 4 | 3.94 | 14.29 |
| | 200 | 8699.09 | 10803.73 | 3.95 | 3 | 7 | 2.55 | 5.90 | 12.13 | 24.55 | 3 | 2.55 | 5.90 |
| Average | | 1342.49 | 2894.03 | 3.98 | 169(sum) | 21(sum) | 0.35 | 1.17 | 4.54 | 6.32 | 144(sum) | 0.68 | 2.45 |

**Table C-7:** Performance of the Solution Approaches for the Resumable Jobs Case When $T_W$ is 30 and $T_U$ is 15

| | | MILP | | | | | | | Heuristic-R | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 3.11 | 3.28 | 3.11 | 10 | 0 | 0.00 | 0.00 | 3.15 | 3.70 | 10 | 0.00 | 0.00 |
| | 20 | 3.14 | 3.37 | 3.14 | 10 | 0 | 0.00 | 0.00 | 3.60 | 4.14 | 10 | 0.00 | 0.00 |
| | 50 | 3.13 | 3.33 | 3.13 | 10 | 0 | 0.00 | 0.00 | 3.55 | 4.08 | 10 | 0.00 | 0.00 |
| | 100 | 3.20 | 3.25 | 3.20 | 10 | 0 | 0.00 | 0.00 | 3.94 | 4.81 | 10 | 0.00 | 0.00 |
| | 200 | 3.20 | 3.32 | 3.20 | 10 | 0 | 0.00 | 0.00 | 4.83 | 7.34 | 8 | 0.03 | 0.16 |
| 3 | 10 | 3.02 | 3.20 | 3.02 | 10 | 0 | 0.00 | 0.00 | 3.44 | 4.08 | 10 | 0.00 | 0.00 |
| | 20 | 3.13 | 3.23 | 3.13 | 10 | 0 | 0.00 | 0.00 | 3.62 | 4.05 | 10 | 0.00 | 0.00 |
| | 50 | 3.21 | 3.29 | 3.21 | 10 | 0 | 0.00 | 0.00 | 3.69 | 4.35 | 9 | 0.10 | 1.03 |
| | 100 | 3.25 | 3.76 | 3.25 | 10 | 0 | 0.00 | 0.00 | 4.01 | 4.81 | 7 | 0.15 | 0.52 |
| | 200 | 45.76 | 474.90 | 4.07 | 10 | 0 | 0.00 | 0.00 | 5.91 | 7.82 | 9 | 0.03 | 0.28 |
| 5 | 20 | 3.12 | 3.28 | 3.12 | 10 | 0 | 0.00 | 0.00 | 3.64 | 4.57 | 10 | 0.00 | 0.00 |
| | 50 | 3.74 | 5.46 | 3.74 | 10 | 0 | 0.00 | 0.00 | 3.85 | 4.62 | 8 | 0.41 | 2.13 |
| | 100 | 54.86 | 339.97 | 3.71 | 10 | 0 | 0.00 | 0.00 | 4.52 | 5.33 | 7 | 2.38 | 21.33 |
| | 200 | 4200.87 | 10804.06 | 3.47 | 7 | 3 | 0.11 | 0.52 | 9.95 | 18.85 | 2 | 0.42 | 0.64 |
| 10 | 50 | 54.21 | 359.46 | 6.84 | 10 | 0 | 0.00 | 0.00 | 3.71 | 4.27 | 9 | 1.00 | 10.00 |
| | 100 | 2355.26 | 10803.47 | 3.46 | 8 | 2 | 0.88 | 4.48 | 4.49 | 5.24 | 6 | 1.91 | 5.56 |
| | 200 | 4533.41 | 10803.83 | 11.23 | 5 | 5 | 0.44 | 1.37 | 21.11 | 29.86 | 1 | 1.39 | 2.00 |
| 20 | 100 | 5528.68 | 10803.51 | 3.58 | 4 | 6 | 2.75 | 9.29 | 6.11 | 8.47 | 4 | 2.75 | 9.29 |
| | 200 | 8699.09 | 10803.71 | 3.95 | 2 | 8 | 2.60 | 6.06 | 14.58 | 30.08 | 1 | 4.14 | 7.81 |
| Average | | 1342.49 | 2906.93 | 3.98 | 166(sum) | 24(sum) | 0.36 | 1.14 | 5.88 | 8.45 | 141(sum) | 0.77 | 3.20 |

**Table C-8**: Performance of the Solution Approaches for the Resumable Jobs Case When $T_W$ is 30 and $T_U$ is 30

| | | MILP | | | | | | | Heuristic-R | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 3.22 | 3.52 | 3.22 | 10 | 0 | 0.00 | 0.00 | 3.23 | 3.80 | 10 | 0.00 | 0.00 |
| | 20 | 3.33 | 3.47 | 3.33 | 10 | 0 | 0.00 | 0.00 | 3.65 | 4.41 | 10 | 0.00 | 0.00 |
| | 50 | 3.24 | 3.34 | 3.24 | 10 | 0 | 0.00 | 0.00 | 3.89 | 4.64 | 10 | 0.00 | 0.00 |
| | 100 | 3.32 | 3.39 | 3.32 | 10 | 0 | 0.00 | 0.00 | 4.02 | 5.42 | 10 | 0.00 | 0.00 |
| | 200 | 3.33 | 3.47 | 3.33 | 10 | 0 | 0.00 | 0.00 | 5.04 | 7.92 | 8 | 0.02 | 0.12 |
| 3 | 10 | 3.15 | 3.35 | 3.15 | 10 | 0 | 0.00 | 0.00 | 3.50 | 3.84 | 10 | 0.00 | 0.00 |
| | 20 | 3.34 | 3.68 | 3.34 | 10 | 0 | 0.00 | 0.00 | 3.83 | 4.18 | 10 | 0.00 | 0.00 |
| | 50 | 3.23 | 3.48 | 3.23 | 10 | 0 | 0.00 | 0.00 | 3.90 | 4.44 | 9 | 0.08 | 0.79 |
| | 100 | 3.45 | 3.70 | 3.45 | 10 | 0 | 0.00 | 0.00 | 4.12 | 5.18 | 8 | 0.08 | 0.39 |
| | 200 | 47.42 | 402.28 | 3.44 | 10 | 0 | 0.00 | 0.00 | 6.58 | 11.09 | 9 | 0.02 | 0.21 |
| 5 | 20 | 3.18 | 3.38 | 3.18 | 10 | 0 | 0.00 | 0.00 | 3.72 | 4.47 | 10 | 0.00 | 0.00 |
| | 50 | 3.91 | 6.13 | 3.91 | 10 | 0 | 0.00 | 0.00 | 4.16 | 4.72 | 7 | 0.47 | 1.61 |
| | 100 | 55.76 | 205.90 | 4.24 | 10 | 0 | 0.00 | 0.00 | 4.78 | 5.62 | 7 | 3.64 | 34.44 |
| | 200 | 4325.52 | 10804.07 | 4.24 | 6 | 4 | 0.11 | 0.40 | 10.99 | 19.78 | 3 | 0.31 | 0.67 |
| 10 | 50 | 65.81 | 359.11 | 5.02 | 10 | 0 | 0.00 | 0.00 | 3.85 | 4.32 | 9 | 1.00 | 10.00 |
| | 100 | 2435.55 | 10803.47 | 3.17 | 8 | 2 | 0.86 | 4.50 | 4.60 | 5.50 | 8 | 0.86 | 4.50 |
| | 200 | 4849.66 | 10805.56 | 3.98 | 6 | 4 | 0.35 | 1.10 | 21.86 | 25.97 | 1 | 1.08 | 1.54 |
| 20 | 100 | 6632.40 | 10803.65 | 3.94 | 4 | 6 | 2.75 | 9.29 | 5.79 | 7.77 | 4 | 2.75 | 9.29 |
| | 200 | 8757.19 | 10803.88 | 3.59 | 3 | 7 | 2.65 | 6.06 | 12.60 | 20.85 | 3 | 2.65 | 6.06 |
| Average | | 1431.89 | 2896.25 | 3.60 | 167(sum) | 23(sum) | 0.35 | 1.12 | 6.01 | 8.10 | 146(sum) | 0.68 | 3.66 |

**Table C-9:** Performance of the Solution Approaches for the Resumable Jobs Case When $T_W$ is 50 and $T_U$ is 25

| | | MILP | | | | | | | Heuristic-R | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 3.04 | 3.25 | 3.04 | 10 | 0 | 0.00 | 0.00 | 3.13 | 3.66 | 10 | 0.00 | 0.00 |
| | 20 | 3.08 | 3.18 | 3.08 | 10 | 0 | 0.00 | 0.00 | 3.17 | 3.53 | 10 | 0.00 | 0.00 |
| | 50 | 3.12 | 3.24 | 3.12 | 10 | 0 | 0.00 | 0.00 | 3.14 | 3.35 | 10 | 0.00 | 0.00 |
| | 100 | 3.13 | 3.23 | 3.13 | 10 | 0 | 0.00 | 0.00 | 3.30 | 3.79 | 8 | 0.03 | 0.19 |
| | 200 | 3.21 | 3.32 | 3.21 | 10 | 0 | 0.00 | 0.00 | 6.03 | 8.76 | 3 | 0.04 | 0.17 |
| 3 | 10 | 3.07 | 3.20 | 3.07 | 10 | 0 | 0.00 | 0.00 | 3.12 | 3.72 | 10 | 0.00 | 0.00 |
| | 20 | 3.09 | 3.18 | 3.09 | 10 | 0 | 0.00 | 0.00 | 3.23 | 3.49 | 10 | 0.00 | 0.00 |
| | 50 | 3.23 | 3.38 | 3.23 | 10 | 0 | 0.00 | 0.00 | 3.19 | 3.42 | 7 | 0.07 | 0.29 |
| | 100 | 3.23 | 3.36 | 3.23 | 10 | 0 | 0.00 | 0.00 | 3.84 | 4.74 | 5 | 0.13 | 0.77 |
| | 200 | 5.65 | 25.34 | 5.61 | 10 | 0 | 0.00 | 0.00 | 7.71 | 10.30 | 2 | 0.08 | 0.21 |
| 5 | 20 | 3.21 | 3.56 | 3.21 | 10 | 0 | 0.00 | 0.00 | 3.08 | 3.64 | 8 | 0.38 | 2.50 |
| | 50 | 3.53 | 3.96 | 3.53 | 10 | 0 | 0.00 | 0.00 | 7.96 | 50.58 | 7 | 0.28 | 1.63 |
| | 100 | 5.53 | 9.69 | 5.44 | 10 | 0 | 0.00 | 0.00 | 4.22 | 5.31 | 1 | 0.37 | 0.77 |
| | 200 | 20.13 | 57.89 | 10.65 | 10 | 0 | 0.00 | 0.00 | 13.35 | 18.98 | 1 | 0.26 | 0.71 |
| 10 | 50 | 32.22 | 284.61 | 4.23 | 10 | 0 | 0.00 | 0.00 | 3.40 | 3.55 | 7 | 0.61 | 2.50 |
| | 100 | 307.99 | 1274.29 | 192.22 | 10 | 0 | 0.00 | 0.00 | 6.02 | 7.49 | 1 | 1.30 | 2.97 |
| | 200 | 7094.21 | 10803.61 | 1601.05 | 6 | 4 | 0.29 | 1.01 | 31.00 | 48.21 | 0 | 2.64 | 13.57 |
| 20 | 100 | 1713.46 | 6883.57 | 718.62 | 10 | 0 | 0.00 | 0.00 | 11.08 | 15.25 | 5 | 2.14 | 4.55 |
| | 200 | 10803.67 | 10804.05 | 39.58 | 0 | 10 | 3.25 | 4.03 | 61.63 | 93.22 | 0 | 3.74 | 6.07 |
| Average | | 1053.57 | 1588.42 | 137.49 | 176(sum) | 14(sum) | 0.19 | 0.27 | 9.56 | 15.53 | 105(sum) | 0.64 | 1.94 |

**Table C-10:** Performance of the Solution Approaches for the Resumable Jobs Case When $T_W$ is 50 and $T_U$ is 50

| | | MILP | | | | | | | Heuristic-R | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 3.02 | 3.20 | 3.02 | 10 | 0 | 0.00 | 0.00 | 3.11 | 3.26 | 10 | 0.00 | 0.00 |
| | 20 | 3.10 | 3.30 | 3.10 | 10 | 0 | 0.00 | 0.00 | 3.05 | 3.22 | 10 | 0.00 | 0.00 |
| | 50 | 3.10 | 3.26 | 3.10 | 10 | 0 | 0.00 | 0.00 | 3.27 | 3.72 | 10 | 0.00 | 0.00 |
| | 100 | 3.19 | 3.37 | 3.19 | 10 | 0 | 0.00 | 0.00 | 3.48 | 4.35 | 9 | 0.01 | 0.14 |
| | 200 | 3.22 | 3.34 | 3.22 | 10 | 0 | 0.00 | 0.00 | 7.52 | 12.36 | 3 | 0.03 | 0.15 |
| 3 | 10 | 3.06 | 3.17 | 3.06 | 10 | 0 | 0.00 | 0.00 | 3.19 | 3.67 | 10 | 0.00 | 0.00 |
| | 20 | 3.11 | 3.25 | 3.11 | 10 | 0 | 0.00 | 0.00 | 3.09 | 3.24 | 10 | 0.00 | 0.00 |
| | 50 | 3.21 | 3.38 | 3.21 | 10 | 0 | 0.00 | 0.00 | 3.26 | 3.50 | 7 | 0.07 | 0.32 |
| | 100 | 3.21 | 3.30 | 3.21 | 10 | 0 | 0.00 | 0.00 | 3.81 | 5.14 | 6 | 0.09 | 0.49 |
| | 200 | 6.14 | 30.60 | 6.11 | 10 | 0 | 0.00 | 0.00 | 11.58 | 19.64 | 1 | 0.07 | 0.16 |
| 5 | 20 | 3.18 | 3.34 | 3.18 | 10 | 0 | 0.00 | 0.00 | 3.10 | 3.65 | 8 | 0.29 | 1.90 |
| | 50 | 3.37 | 3.67 | 3.37 | 10 | 0 | 0.00 | 0.00 | 3.25 | 3.67 | 6 | 0.22 | 0.85 |
| | 100 | 4.77 | 7.05 | 4.77 | 10 | 0 | 0.00 | 0.00 | 4.57 | 6.30 | 1 | 0.31 | 0.58 |
| | 200 | 19.50 | 64.43 | 7.16 | 10 | 0 | 0.00 | 0.00 | 14.36 | 21.54 | 1 | 0.21 | 0.54 |
| 10 | 50 | 20.64 | 167.86 | 4.36 | 10 | 0 | 0.00 | 0.00 | 3.37 | 4.26 | 6 | 0.81 | 2.50 |
| | 100 | 104.43 | 366.34 | 19.86 | 10 | 0 | 0.00 | 0.00 | 7.53 | 9.14 | 1 | 1.12 | 3.17 |
| | 200 | 6526.66 | 10805.79 | 249.33 | 6 | 4 | 0.22 | 0.76 | 34.92 | 66.75 | 0 | 0.95 | 1.75 |
| 20 | 100 | 1927.98 | 10803.28 | 61.87 | 9 | 1 | 0.43 | 4.35 | 11.53 | 16.93 | 3 | 3.01 | 4.55 |
| | 200 | 10803.63 | 10803.77 | 159.62 | 0 | 10 | 3.16 | 3.81 | 59.17 | 96.41 | 0 | 3.16 | 3.81 |
| Average | | 1023.61 | 1741.35 | 28.83 | 175(sum) | 15(sum) | 0.20 | 0.47 | 9.85 | 15.30 | 102(sum) | 0.55 | 1.10 |

**Table C-11:** Performance of the Solution Approaches for the Resumable Jobs Case When $T_W$ is 100 and $T_U$ is 20

| | | MILP | | | | | | | Heuristic-R | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 3.10 | 3.30 | 3.10 | 10 | 0 | 0.00 | 0.00 | 3.03 | 3.27 | 10 | 0.00 | 0.00 |
| | 20 | 3.21 | 3.34 | 3.21 | 10 | 0 | 0.00 | 0.00 | 3.13 | 3.38 | 10 | 0.00 | 0.00 |
| | 50 | 3.14 | 3.23 | 3.14 | 10 | 0 | 0.00 | 0.00 | 3.12 | 3.60 | 10 | 0.00 | 0.00 |
| | 100 | 3.18 | 3.39 | 3.18 | 10 | 0 | 0.00 | 0.00 | 3.28 | 3.60 | 8 | 0.02 | 0.16 |
| | 200 | 3.24 | 3.31 | 3.24 | 10 | 0 | 0.00 | 0.00 | 5.08 | 7.36 | 4 | 0.05 | 0.21 |
| 3 | 10 | 3.21 | 3.39 | 3.21 | 10 | 0 | 0.00 | 0.00 | 3.08 | 3.61 | 10 | 0.00 | 0.00 |
| | 20 | 3.18 | 3.42 | 3.18 | 10 | 0 | 0.00 | 0.00 | 3.12 | 3.40 | 10 | 0.00 | 0.00 |
| | 50 | 3.23 | 3.38 | 3.23 | 10 | 0 | 0.00 | 0.00 | 3.18 | 3.63 | 7 | 0.11 | 0.52 |
| | 100 | 3.30 | 3.45 | 3.30 | 10 | 0 | 0.00 | 0.00 | 3.61 | 4.41 | 7 | 0.14 | 0.96 |
| | 200 | 3.53 | 3.87 | 3.53 | 10 | 0 | 0.00 | 0.00 | 7.09 | 10.12 | 1 | 0.11 | 0.22 |
| 5 | 20 | 3.20 | 3.31 | 3.20 | 10 | 0 | 0.00 | 0.00 | 3.09 | 3.74 | 8 | 0.56 | 3.64 |
| | 50 | 3.53 | 3.97 | 3.53 | 10 | 0 | 0.00 | 0.00 | 3.24 | 3.57 | 7 | 0.34 | 1.95 |
| | 100 | 4.62 | 6.88 | 4.62 | 10 | 0 | 0.00 | 0.00 | 4.04 | 5.41 | 1 | 0.51 | 1.07 |
| | 200 | 19.09 | 60.06 | 8.09 | 10 | 0 | 0.00 | 0.00 | 12.79 | 19.90 | 1 | 0.36 | 0.90 |
| 10 | 50 | 4.62 | 6.20 | 4.62 | 10 | 0 | 0.00 | 0.00 | 3.37 | 4.10 | 6 | 1.16 | 5.00 |
| | 100 | 60.39 | 182.57 | 20.10 | 10 | 0 | 0.00 | 0.00 | 5.35 | 6.50 | 4 | 0.78 | 1.56 |
| | 200 | 9099.13 | 10803.68 | 1052.42 | 3 | 7 | 0.69 | 1.26 | 36.79 | 75.62 | 0 | 1.91 | 3.24 |
| 20 | 100 | 1463.29 | 4655.04 | 49.26 | 10 | 0 | 0.00 | 0.00 | 11.49 | 15.89 | 5 | 2.14 | 4.55 |
| | 200 | 10803.67 | 10804.06 | 85.14 | 0 | 10 | 3.18 | 3.96 | 58.44 | 84.15 | 0 | 3.68 | 6.16 |
| Average | | 1131.25 | 1397.89 | 66.49 | 173(sum) | 17(sum) | 0.20 | 0.27 | 9.28 | 13.96 | 109(sum) | 0.62 | 1.59 |

**Table C-12:** Performance of the Solution Approaches for the Resumable Jobs Case When $T_W$ is 100 and $T_U$ is 50

| | | MILP | | | | | | | Heuristic-R | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 3.19 | 3.42 | 3.19 | 10 | 0 | 0.00 | 0.00 | 3.02 | 3.39 | 10 | 0.00 | 0.00 |
| | 20 | 3.27 | 3.37 | 3.27 | 10 | 0 | 0.00 | 0.00 | 3.04 | 3.17 | 10 | 0.00 | 0.00 |
| | 50 | 3.30 | 3.54 | 3.30 | 10 | 0 | 0.00 | 0.00 | 3.13 | 3.66 | 10 | 0.00 | 0.00 |
| | 100 | 3.30 | 3.55 | 3.30 | 10 | 0 | 0.00 | 0.00 | 3.41 | 4.38 | 8 | 0.03 | 0.19 |
| | 200 | 3.37 | 3.62 | 3.37 | 10 | 0 | 0.00 | 0.00 | 6.18 | 12.95 | 3 | 0.04 | 0.10 |
| 3 | 10 | 3.26 | 3.39 | 3.26 | 10 | 0 | 0.00 | 0.00 | 3.09 | 3.61 | 10 | 0.00 | 0.00 |
| | 20 | 3.32 | 3.70 | 3.32 | 10 | 0 | 0.00 | 0.00 | 3.09 | 3.19 | 10 | 0.00 | 0.00 |
| | 50 | 3.37 | 3.60 | 3.37 | 10 | 0 | 0.00 | 0.00 | 3.14 | 3.47 | 7 | 0.09 | 0.42 |
| | 100 | 3.42 | 3.66 | 3.42 | 10 | 0 | 0.00 | 0.00 | 3.76 | 4.50 | 5 | 0.13 | 0.77 |
| | 200 | 3.70 | 4.04 | 3.70 | 10 | 0 | 0.00 | 0.00 | 7.88 | 14.95 | 3 | 0.08 | 0.22 |
| 5 | 20 | 3.25 | 3.36 | 3.25 | 10 | 0 | 0.00 | 0.00 | 3.09 | 3.93 | 8 | 0.56 | 3.64 |
| | 50 | 3.63 | 4.24 | 3.63 | 10 | 0 | 0.00 | 0.00 | 3.25 | 3.72 | 5 | 0.39 | 1.63 |
| | 100 | 4.60 | 6.02 | 4.60 | 10 | 0 | 0.00 | 0.00 | 4.20 | 5.67 | 0 | 0.48 | 0.86 |
| | 200 | 18.64 | 59.09 | 7.40 | 10 | 0 | 0.00 | 0.00 | 13.63 | 23.15 | 1 | 0.29 | 0.73 |
| 10 | 50 | 5.34 | 9.61 | 5.34 | 10 | 0 | 0.00 | 0.00 | 3.34 | 3.92 | 6 | 1.16 | 5.00 |
| | 100 | 82.93 | 267.68 | 30.29 | 10 | 0 | 0.00 | 0.00 | 5.29 | 6.70 | 5 | 0.66 | 1.56 |
| | 200 | 5783.51 | 10803.90 | 847.42 | 7 | 3 | 0.25 | 0.99 | 40.65 | 69.78 | 0 | 1.43 | 2.63 |
| 20 | 100 | 1491.41 | 4558.51 | 278.06 | 10 | 0 | 0.00 | 0.00 | 11.88 | 17.79 | 4 | 2.58 | 4.55 |
| | 200 | 10803.80 | 10804.21 | 40.14 | 0 | 10 | 3.15 | 3.91 | 57.28 | 84.18 | 0 | 3.64 | 6.10 |
| Average | | 959.51 | 1397.50 | 65.98 | 177(sum) | 13(sum) | 0.18 | 0.26 | 9.60 | 14.53 | 105(sum) | 0.61 | 1.50 |

**Table C-13:** Performance of the Solution Approaches for the Resumable Jobs Case When $T_W$ is 100 and $T_U$ is 100

| | | MILP | | | | | | | Heuristic-R | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 2.70 | 2.80 | 2.70 | 10 | 0 | 0.00 | 0.00 | 3.00 | 3.30 | 10 | 0.00 | 0.00 |
| | 20 | 2.70 | 2.77 | 2.70 | 10 | 0 | 0.00 | 0.00 | 3.04 | 3.29 | 10 | 0.00 | 0.00 |
| | 50 | 2.72 | 2.80 | 2.72 | 10 | 0 | 0.00 | 0.00 | 3.10 | 3.47 | 10 | 0.00 | 0.00 |
| | 100 | 2.80 | 2.91 | 2.80 | 10 | 0 | 0.00 | 0.00 | 3.32 | 3.94 | 9 | 0.01 | 0.15 |
| | 200 | 2.83 | 2.96 | 2.83 | 10 | 0 | 0.00 | 0.00 | 6.19 | 11.09 | 2 | 0.03 | 0.12 |
| 3 | 10 | 2.68 | 2.80 | 2.68 | 10 | 0 | 0.00 | 0.00 | 3.06 | 3.42 | 10 | 0.00 | 0.00 |
| | 20 | 2.75 | 2.87 | 2.75 | 10 | 0 | 0.00 | 0.00 | 3.12 | 3.62 | 10 | 0.00 | 0.00 |
| | 50 | 2.81 | 3.00 | 2.81 | 10 | 0 | 0.00 | 0.00 | 3.15 | 3.46 | 7 | 0.05 | 0.22 |
| | 100 | 2.85 | 3.00 | 2.85 | 10 | 0 | 0.00 | 0.00 | 3.58 | 4.95 | 5 | 0.10 | 0.59 |
| | 200 | 3.11 | 3.41 | 3.11 | 10 | 0 | 0.00 | 0.00 | 9.38 | 21.54 | 1 | 0.08 | 0.16 |
| 5 | 20 | 2.73 | 2.91 | 2.73 | 10 | 0 | 0.00 | 0.00 | 3.03 | 3.37 | 8 | 0.56 | 3.64 |
| | 50 | 2.93 | 3.22 | 2.93 | 10 | 0 | 0.00 | 0.00 | 3.27 | 3.49 | 6 | 0.26 | 1.28 |
| | 100 | 4.19 | 5.51 | 4.19 | 10 | 0 | 0.00 | 0.00 | 4.30 | 5.76 | 0 | 0.44 | 0.89 |
| | 200 | 19.02 | 70.96 | 6.54 | 10 | 0 | 0.00 | 0.00 | 14.38 | 22.59 | 1 | 0.22 | 0.56 |
| 10 | 50 | 4.21 | 5.52 | 4.21 | 10 | 0 | 0.00 | 0.00 | 3.34 | 3.76 | 7 | 0.72 | 2.50 |
| | 100 | 64.97 | 272.82 | 36.44 | 10 | 0 | 0.00 | 0.00 | 5.33 | 7.51 | 6 | 0.54 | 1.56 |
| | 200 | 7019.57 | 10803.70 | 384.98 | 6 | 4 | 0.30 | 0.97 | 43.80 | 69.03 | 0 | 1.19 | 2.20 |
| 20 | 100 | 1603.09 | 4551.20 | 278.22 | 10 | 0 | 0.00 | 0.00 | 11.49 | 16.08 | 6 | 1.71 | 4.55 |
| | 200 | 10803.21 | 10803.30 | 973.53 | 0 | 10 | 3.21 | 3.93 | 57.44 | 82.21 | 0 | 3.70 | 6.08 |
| Average | | 1029.05 | 1397.29 | 90.62 | 176(sum) | 14(sum) | 0.18 | 0.26 | 9.86 | 14.52 | 108(sum) | 0.51 | 1.29 |

**Table C-14:** Performance of the Solution Approaches for the Resumable Jobs Case When $T_W$ is 150 and $T_U$ is 30

| | | MILP | | | | | | | Heuristic-R | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 3.01 | 3.19 | 3.01 | 10 | 0 | 0.00 | 0.00 | 3.06 | 3.36 | 10 | 0.00 | 0.00 |
| | 20 | 3.06 | 3.20 | 3.06 | 10 | 0 | 0.00 | 0.00 | 3.09 | 3.44 | 10 | 0.00 | 0.00 |
| | 50 | 3.04 | 3.13 | 3.04 | 10 | 0 | 0.00 | 0.00 | 3.12 | 3.53 | 10 | 0.00 | 0.00 |
| | 100 | 3.13 | 3.25 | 3.13 | 10 | 0 | 0.00 | 0.00 | 3.34 | 3.96 | 8 | 0.02 | 0.15 |
| | 200 | 3.21 | 3.38 | 3.21 | 10 | 0 | 0.00 | 0.00 | 5.07 | 8.08 | 5 | 0.04 | 0.21 |
| 3 | 10 | 3.02 | 3.11 | 3.02 | 10 | 0 | 0.00 | 0.00 | 3.07 | 3.45 | 10 | 0.00 | 0.00 |
| | 20 | 3.10 | 3.26 | 3.10 | 10 | 0 | 0.00 | 0.00 | 3.10 | 3.31 | 10 | 0.00 | 0.00 |
| | 50 | 3.09 | 3.24 | 3.09 | 10 | 0 | 0.00 | 0.00 | 3.15 | 3.46 | 8 | 0.09 | 0.52 |
| | 100 | 3.20 | 3.31 | 3.20 | 10 | 0 | 0.00 | 0.00 | 3.71 | 4.47 | 6 | 0.10 | 0.49 |
| | 200 | 3.48 | 3.98 | 3.48 | 10 | 0 | 0.00 | 0.00 | 7.41 | 10.14 | 1 | 0.13 | 0.22 |
| 5 | 20 | 3.04 | 3.12 | 3.04 | 10 | 0 | 0.00 | 0.00 | 3.06 | 3.74 | 8 | 0.56 | 3.64 |
| | 50 | 3.25 | 3.60 | 3.25 | 10 | 0 | 0.00 | 0.00 | 3.28 | 3.51 | 8 | 0.16 | 0.83 |
| | 100 | 4.83 | 7.41 | 4.83 | 10 | 0 | 0.00 | 0.00 | 4.15 | 5.75 | 0 | 0.60 | 1.11 |
| | 200 | 13.87 | 25.08 | 9.19 | 10 | 0 | 0.00 | 0.00 | 13.86 | 20.36 | 0 | 0.31 | 0.46 |
| 10 | 50 | 4.73 | 8.38 | 4.23 | 10 | 0 | 0.00 | 0.00 | 3.39 | 3.88 | 7 | 0.97 | 5.00 |
| | 100 | 105.31 | 632.93 | 43.43 | 10 | 0 | 0.00 | 0.00 | 5.30 | 7.54 | 4 | 0.78 | 1.56 |
| | 200 | 6397.10 | 10803.76 | 583.85 | 6 | 4 | 0.31 | 0.89 | 59.13 | 88.89 | 0 | 2.03 | 2.95 |
| 20 | 100 | 1964.19 | 6063.15 | 239.22 | 10 | 0 | 0.00 | 0.00 | 11.43 | 17.06 | 6 | 1.71 | 4.55 |
| | 200 | 10803.59 | 10803.72 | 23.67 | 0 | 10 | 3.16 | 3.68 | 56.98 | 79.52 | 0 | 3.65 | 6.15 |
| Average | | 1017.43 | 1493.91 | 49.79 | 176(sum) | 14(sum) | 0.18 | 0.24 | 10.46 | 14.60 | 111(sum) | 0.59 | 1.47 |

**Table C-15:** Performance of the Solution Approaches for the Resumable Jobs Case When $T_W$ is 150 and $T_U$ is 75

| | | MILP | | | | | | | Heuristic-R | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 2.73 | 3.20 | 2.73 | 10 | 0 | 0.00 | 0.00 | 3.03 | 3.42 | 10 | 0.00 | 0.00 |
| | 20 | 2.71 | 2.82 | 2.71 | 10 | 0 | 0.00 | 0.00 | 3.11 | 3.44 | 10 | 0.00 | 0.00 |
| | 50 | 2.76 | 3.46 | 2.76 | 10 | 0 | 0.00 | 0.00 | 3.16 | 3.60 | 10 | 0.00 | 0.00 |
| | 100 | 2.77 | 2.94 | 2.77 | 10 | 0 | 0.00 | 0.00 | 3.36 | 3.85 | 9 | 0.01 | 0.12 |
| | 200 | 2.81 | 3.01 | 2.81 | 10 | 0 | 0.00 | 0.00 | 5.53 | 10.64 | 3 | 0.04 | 0.10 |
| 3 | 10 | 2.68 | 2.81 | 2.68 | 10 | 0 | 0.00 | 0.00 | 3.09 | 3.61 | 10 | 0.00 | 0.00 |
| | 20 | 2.70 | 2.84 | 2.70 | 10 | 0 | 0.00 | 0.00 | 3.13 | 3.24 | 10 | 0.00 | 0.00 |
| | 50 | 2.77 | 2.93 | 2.77 | 10 | 0 | 0.00 | 0.00 | 3.22 | 3.58 | 7 | 0.12 | 0.63 |
| | 100 | 2.76 | 2.88 | 2.76 | 10 | 0 | 0.00 | 0.00 | 3.85 | 5.02 | 5 | 0.14 | 0.80 |
| | 200 | 3.08 | 3.33 | 3.08 | 10 | 0 | 0.00 | 0.00 | 8.56 | 18.15 | 1 | 0.11 | 0.22 |
| 5 | 20 | 2.70 | 2.86 | 2.70 | 10 | 0 | 0.00 | 0.00 | 3.06 | 3.68 | 8 | 0.56 | 3.64 |
| | 50 | 2.86 | 3.05 | 2.86 | 10 | 0 | 0.00 | 0.00 | 3.27 | 3.66 | 8 | 0.16 | 0.83 |
| | 100 | 4.42 | 10.07 | 4.42 | 10 | 0 | 0.00 | 0.00 | 4.20 | 5.76 | 0 | 0.58 | 0.95 |
| | 200 | 15.99 | 37.47 | 10.52 | 10 | 0 | 0.00 | 0.00 | 15.56 | 21.98 | 0 | 0.26 | 0.38 |
| 10 | 50 | 5.46 | 11.85 | 5.18 | 10 | 0 | 0.00 | 0.00 | 3.37 | 4.04 | 7 | 0.97 | 5.00 |
| | 100 | 61.66 | 271.76 | 11.07 | 10 | 0 | 0.00 | 0.00 | 5.27 | 7.29 | 5 | 0.66 | 1.56 |
| | 200 | 7423.67 | 10803.89 | 1477.83 | 6 | 4 | 0.28 | 0.84 | 35.01 | 73.25 | 1 | 0.94 | 2.36 |
| 20 | 100 | 1434.64 | 2768.28 | 388.85 | 10 | 0 | 0.00 | 0.00 | 10.62 | 16.06 | 6 | 1.71 | 4.55 |
| | 200 | 10803.22 | 10803.36 | 94.78 | 0 | 10 | 3.44 | 6.19 | 56.78 | 83.90 | 0 | 3.92 | 6.27 |
| Average | | 1041.18 | 1302.25 | 106.63 | 176(sum) | 14(sum) | 0.20 | 0.37 | 9.33 | 14.64 | 110(sum) | 0.53 | 1.44 |

**Table C-16:** Performance of the Solution Approaches for the Resumable Jobs Case When $T_W$ is 150 and $T_U$ is 150

| | | MILP | | | | | | | Heuristic-R | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *n* | Average CPU time | Max. CPU time | Average CPU time for BI | Number of optimum solutions obtained | Number of non optimum integer solutions obtained | Average percent error | Max. percent error | Average CPU time | Max. CPU time | Number of optimum solutions obtained | Average percent error | Max. percent error |
| 2 | 10 | 3.05 | 3.19 | 3.05 | 10 | 0 | 0.00 | 0.00 | 3.03 | 3.48 | 10 | 0.00 | 0.00 |
| | 20 | 3.07 | 3.18 | 3.07 | 10 | 0 | 0.00 | 0.00 | 3.09 | 3.29 | 10 | 0.00 | 0.00 |
| | 50 | 3.11 | 3.22 | 3.11 | 10 | 0 | 0.00 | 0.00 | 3.10 | 3.53 | 10 | 0.00 | 0.00 |
| | 100 | 3.15 | 3.27 | 3.15 | 10 | 0 | 0.00 | 0.00 | 3.34 | 3.91 | 8 | 0.02 | 0.09 |
| | 200 | 3.24 | 3.37 | 3.24 | 10 | 0 | 0.00 | 0.00 | 5.81 | 11.51 | 2 | 0.04 | 0.13 |
| 3 | 10 | 3.06 | 3.20 | 3.06 | 10 | 0 | 0.00 | 0.00 | 3.05 | 3.45 | 10 | 0.00 | 0.00 |
| | 20 | 3.08 | 3.22 | 3.08 | 10 | 0 | 0.00 | 0.00 | 3.09 | 3.19 | 10 | 0.00 | 0.00 |
| | 50 | 3.17 | 3.36 | 3.17 | 10 | 0 | 0.00 | 0.00 | 3.17 | 3.71 | 7 | 0.09 | 0.32 |
| | 100 | 3.27 | 3.55 | 3.27 | 10 | 0 | 0.00 | 0.00 | 3.80 | 4.60 | 6 | 0.11 | 0.62 |
| | 200 | 3.46 | 3.74 | 3.46 | 10 | 0 | 0.00 | 0.00 | 9.61 | 16.55 | 1 | 0.09 | 0.20 |
| 5 | 20 | 3.06 | 3.19 | 3.06 | 10 | 0 | 0.00 | 0.00 | 3.05 | 3.46 | 8 | 0.56 | 3.64 |
| | 50 | 3.28 | 3.61 | 3.28 | 10 | 0 | 0.00 | 0.00 | 3.23 | 3.40 | 8 | 0.16 | 0.83 |
| | 100 | 4.81 | 6.80 | 4.78 | 10 | 0 | 0.00 | 0.00 | 4.58 | 6.39 | 1 | 0.39 | 0.77 |
| | 200 | 13.75 | 36.59 | 9.02 | 10 | 0 | 0.00 | 0.00 | 17.30 | 25.25 | 0 | 0.22 | 0.49 |
| 10 | 50 | 5.37 | 11.99 | 5.26 | 10 | 0 | 0.00 | 0.00 | 3.35 | 4.13 | 6 | 1.16 | 5.00 |
| | 100 | 62.63 | 137.95 | 14.88 | 10 | 0 | 0.00 | 0.00 | 5.27 | 7.18 | 4 | 0.78 | 1.56 |
| | 200 | 6108.67 | 10803.86 | 431.43 | 6 | 4 | 0.27 | 1.31 | 32.46 | 59.03 | 1 | 0.86 | 1.76 |
| 20 | 100 | 1437.06 | 2781.78 | 387.68 | 10 | 0 | 0.00 | 0.00 | 10.81 | 15.78 | 6 | 1.71 | 4.55 |
| | 200 | 10803.62 | 10803.72 | 31.13 | 0 | 10 | 3.43 | 6.19 | 56.26 | 79.64 | 0 | 3.67 | 6.27 |
| Average | | 972.31 | 1295.94 | 48.53 | 176(sum) | 14(sum) | 0.20 | 0.39 | 9.34 | 13.76 | 108(sum) | 0.52 | 1.38 |

# APPENDIX D

# CURRICULUM VITAE

## PERSONAL INFORMATION

Surname, Name: KURT, Atıl

Nationality: Turkish (TC)

Date and Place of Birth: 01 September 1986, Hekimhan

Marital Status: Single

Phone: +90 312 233 13 74

Email: atilkurt@cankaya.edu.tr

## EDUCATION

| Degree | Institution | Year of Graduation |
|---|---|---|
| BS | Çankaya Univ. Industrial Engineering | 2008 |
| High School | Seyranbagları High school, Ankara | 2003 |

## WORK EXPERIENCE

| Year | Place | Enrollment |
|---|---|---|
| 2009-Present | Cankaya Univ. Department of Industrial Engineering | Specialist |
| 2007 July- August | Arcelik-Refrigerator Plant | Intern Student |
| 2006 July- August | Hidromek Inc. | Intern Student |

## FOREIGN LANGUAGES

Upper Intermediate English

## HOBBIES

Football, Movies, Latin Dances